# Getting the Picture

Robert Akerlof, Richard Holden, and Hongyi Li[*]

May 14, 2025

**Abstract**

People may have all the information needed to draw a conclusion yet—in contrast to standard economic models—they fail to connect the dots. For example, when staring at a picture, someone might be able to identify the color of each "pixel" and yet fail to see what the picture represents (e.g. "it's a smiley face"). We formalize this idea. An agent's task is to learn about a picture's features. Initially, they know the color of each pixel, but not features of larger regions of the picture—thus, they cannot discern what the picture depicts. They add to knowledge by loading existing knowledge into working memory and deducing new features. Importantly, the agent has limited working memory, which bounds their ability to draw conclusions. The model captures many important phenomena, such as multi-stable perception, choice overload, and satisficing. It provides a useful conceptualization of narratives as "big-picture statements." We discuss several potential applications, including to the politics of persuasion.

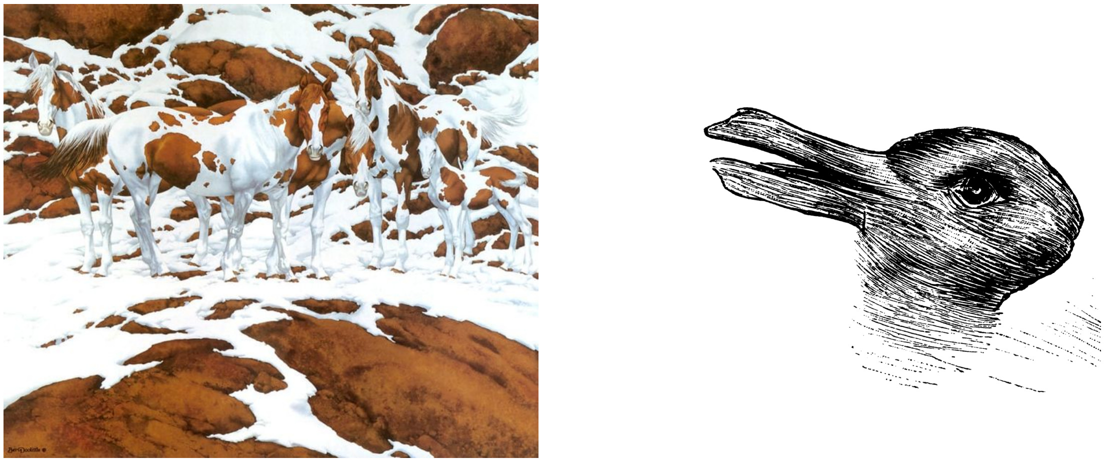
*JEL Classification:* D01, D80, D90.

*Keywords:* Cognition, reasoning, perception, narratives.

[*] Akerlof: UNSW Business School and University of Warwick, r.akerlof@unsw.edu.au. Holden: UNSW Business School, richard.holden@unsw.edu.au. Li: UNSW Business School, hongyi@hongyi.li. We are grateful to Rachel Kranton, Raghav Malhotra, Sharun Mukand, Andrei Shleifer, and Richard Thaler for helpful comments, as well as seminar participants at Harvard, MIT, Brown, Monash, and UNSW. Li gratefully acknowledges financial support from the Australian Research Council Discovery Project DP240103257.

# 1 Introduction

In the early 20th century, a new school of psychology emerged that challenged prevailing notions regarding human perception. The so-called Gestalt theorists, such as Max Wertheimer, realized that there is a fundamental difference between perceiving the *parts* of a picture and perceiving its *whole*. People may struggle to see the big picture. We have all had that "aha" moment where a scene suddenly becomes clear (see Figure 1(a) for an example). In addition to the struggles all people have with seeing the whole, neuroscientists have classified a variety of disorders ("agnosias") such as *face blindness* ("prosopagnosia"), where individual facial features—eyes, nose, and mouth—are recognizable but it is difficult or impossible to put a name to the face. As Gazzaniga et al. (2014) put it, "it's somewhat like going to Legoland and—instead of seeing the integrated percepts of buildings, cars, and monsters—seeing nothing but piles of legos." On the flipside, Gestalt psychologists observed that people may integrate the parts into a whole in multiple ways—a phenomenon known as perceptual rivalry. A famous example is the rabbit-duck illusion (Figure 1(b)), where some people see a rabbit, while others perceive a duck.



(a) Most people do not see all five horses immediately.

(b) The Rabbit-Duck Illusion

Figure 1

The main focus of the Gestalt psychologists was visual perception, but their insights apply more broadly. In many domains, people have all of the information needed to draw a conclusion yet fail to connect the dots. Consider Fermat's Last Theorem, which remained unproven for over three centuries despite the best efforts of mathematicians.[1] This contrasts sharply with standard economic models, which assume that agents have perfect information processing capabilities and can immediately derive all logical consequences of their knowledge.

Understanding how people connect the dots—or fail to connect the dots—is central to understanding economic decision-making. People are flooded with data; but to take action, they need to make sense of it. In other words, they need a big-picture view—which may usefully be termed a "narrative" (see Shiller (2017)). Take the 2008 financial crisis. One narrative took hold regarding rising house prices—that they were driven by fundamentals—rather than an alternative—that there was a speculative frenzy and loose lending standards. The lack of a compelling story can keep people from taking action. This may explain why defaulting people into savings plans tends to increase enrollment (see Madrian and Shea (2001) and Thaler and Benartzi (2004)).

This paper develops a framework for understanding how people analyze pieces of information and reason their way to a big picture. We also aim to understand how a big picture informs an agent's view of the parts.

We consider a model in which an agent is presented with a picture consisting of a grid of black and white pixels. Their task is to identify the features of the picture. We define a feature as any pattern that applies to some region of the picture. For instance, a feature might identify the color of a particular pixel, or it might identify whether the picture depicts a smiley face.

The agent has a knowledge set—to which they may add over time—consisting of fea-

[1] Fermat's Last Theorem was deducible from the rules of arithmetic. Mathematicians had everything they needed to establish the theorem—they had the "pixels." However, for 358 years, it was not clear to mathematicians whether Fermat's conjecture was true; and the proof Andrew Wiles ultimately provided was complex, involving branches of mathematics (such as modular elliptic curves) unknown to Fermat. Thus, "connecting the dots" was a non-trivial process.

tures that they think apply to the picture. The agent initially knows the color of each pixel, but they do not know any features of larger regions of the picture. In this sense, the agent starts with complete information about the picture but no understanding of what it represents (e.g. "it's a smiley-face").

The agent learns about the picture by loading existing knowledge into working memory and drawing conclusions. For instance, if the agent loaded into working memory "the number of white pixels is even" and "the number of white pixels is prime," they would conclude that "there are two white pixels." Each period, the conclusions the agent draws are added to their knowledge set.[2]

Importantly, the agent has limited working memory, which restricts the amount of knowledge they can load into working memory and bounds their ability to draw conclusions. This constraint means, for instance, that the agent cannot immediately deduce all of the picture's features simply by loading their knowledge of every pixel's color into working memory. We assume that the agent is endowed with a *code* that maps features to binary strings (such as "1101" or "1"), and that each feature takes up working-memory space equal to the length of its codeword.[3]

We consider two variants of the model. In the first, the conclusions the agent draws are purely deductive, while in the second, the agent extrapolates. The pure-deduction model yields three key insights. First, the agent may develop only a piecemeal understanding of the picture. They may, like a person with face blindness, recognize various parts of the

[2] We use the term "working memory" in the way that it is employed by computer scientists: to refer to short-term memory used for processing data. Cognitive scientists, however, use this term to refer to a specific brain system involving conscious awareness, which is divided into components (see Gazzaniga et al. (2014)): the phonological loop (for verbal information), the visuospatial sketchpad (for visual and spatial information), and the central executive (which coordinates attention and integrates information across these systems). For unconscious brain systems that temporarily store and manipulate information, cognitive scientists use a separate term, "implicit memory." In our framework, we abstract between these two types of brain systems (conscious and unconscious).

[3] Cognitive scientists and neuroscientists have shown that the brain relies on encoding to manage complex information. For instance, Baddeley and Hitch (1974) suggest that information is compartmentalized and transformed through encoding processes to reduce cognitive load. This idea is supported by findings on chunking (discussed in Section 2.4), where individuals condense data into compact units, and by Levels of Processing Theory (Craik and Lockhart (1972)), which highlights the importance of transforming information to deepen processing and improve memory retention.

picture but fail to integrate them and see the whole.

Second, some conclusions can only be reached in multiple steps. The reason the agent can see more in two steps than one is that the agent can use the first step to “chunk” information: combine knowledge represented by multiple features into a single, less memory-intensive feature. Chunking allows the agent to store more information in working memory—and thereby deduce more. The term “chunk” is drawn from related research in cognitive psychology (see Miller (1956) and Chase and Simon (1973)).

Third, the agent employs both bottom-up and top-down thinking to make deductions. Bottom-up thinking combines small features into larger ones, while top-down thinking uses big-picture features (narratives) to deduce smaller details. The agent uses top-down thinking, rather than bottom-up alone, because narratives offer a memory-efficient way of thinking about the picture.

In the version of the model with extrapolation, the agent adds features to knowledge when they “fit the data” sufficiently well—in the sense that there are relatively few alternative explanations. Extrapolation expands the set of conclusions that can be reached with limited working memory or limited initial information; but it also introduces the possibility of mistakes.

Extrapolation makes sense of the phenomenon of perceptual rivalry. To see why, consider the rabbit-duck illusion (Figure 1(b)). Depending upon which part of the picture the agent processes first, the agent can get into one of two steady states. In one, the agent extrapolates to a “rabbit” interpretation of the full picture and a “rabbit ears” interpretation of the left-hand side. “Rabbit ears” and “rabbit” mutually reinforce each other. For instance, the presence of “rabbit ears” in working memory blocks the adoption of alternative explanations to “rabbit”—such as “duck.” In the other steady state, the agent adopts a “duck” interpretation of the full picture and a “duck bill” interpretation of the left-hand side. These features, likewise, are mutually reinforcing.

We refer to the interpretation of the parts of the picture (e.g. “rabbit ears” or “duck bill”) as *mental scaffolding*. Mental scaffolding lends stability to the agent’s overall interpretation of the picture—or narrative. When the scaffolding is weak, rather than fixing on a single perception, the agent may cycle between perceptions such as “rabbit” and “duck.”

In most of the paper, we focus on settings where agents draw conclusions without acting upon them. However, we consider a straightforward extension where the agent faces a choice. To choose, the agent must draw a conclusion about which option yields the most utility. We show how limited working memory affects the agent’s perception of their options and thus their decision-making. This extension captures a variety of well known phenomena such as choice overload, selective attention to attributes (e.g. price or quality), satisficing, and the role of defaults.

The model offers new insights into the persuasion problem. In standard economic models of persuasion (e.g. Kamenica and Gentzkow (2011)), persuasion involves controlling the information the agent receives. In our framework, the agent’s beliefs also depend upon how they *interpret* that information, which opens up new persuasion channels. For instance, it matters whether good news is presented before bad news or vice-versa, as this may affect which narrative the decision-maker ultimately adopts. Persuasion may also involve suggesting a narrative. Here, we have in mind that suggesting narratives involves influencing what the agent attends to (i.e. the narratives the agent considers adopting). One application is to the political process, where our model makes sense of why it is important to maintain “control of the narrative” and “get ahead” of damaging stories.

The most persuasive narratives, moreover, tend to be *simple* (i.e. features with short codewords). Simple narratives conserve working memory; consequently, they are easy to employ. In fact, simple narratives that are *false* may be adopted over complex narratives that are *true*. Politicians especially understand the importance of simplicity. Take, for instance, Ronald Reagan’s well known line: “Government is not the solution to our problem, government is the problem.” As the political consultant Frank Luntz puts it:

"the most memorable political language is rarely longer than a sentence."[4]

We conclude the paper with a discussion of potential applications and extensions. We show how the framework can be adapted to examine how an agent might come to perceive the relationship between multiple observations, a process that captures predictive model-building by an agent. Another promising extension is to a setting where two agents communicate. Back-and-forth communication can be fruitful because the agents encode features differently—which leads them to see different things in the same data and share insights with each other.[5]

It is useful to note that, in given applications of the model, it is often clear what types of codes agents will employ.[6] Moreover, once we impose structure on agents' codes, the model delivers sharp predictions about behavior. For instance, in our application to political persuasion, putting structure on voters' codes delivers clear predictions about when politicians will preemptively release damaging information.

There is, of course, a vast literature in economics on bounded rationality, starting with the work of Simon (1955) who pointed to "limits on computational capacity" as an important constraint on "actual human choice." The highly influential line of work on heuristics and biases, initiated by Kahneman and Tversky, suggests that people rely on mental shortcuts to make decisions due to cognitive limitations, though these shortcuts can sometimes lead to systematic errors. This literature, moreover, was strongly influenced by work in cognitive psychology on perception. For instance, they argued that people's perceptions can be skewed by arbitrary reference points—leading to phenomena such as anchoring effects or endowment effects (see Kahneman and Tversky (1974) and Thaler (1980)).

A comparatively recent strand of work by Bordalo, Gennaioli, Shleifer, and coauthors (hereafter BGS et al.) seeks to "get inside people's heads" and explore the consequences

[4] Luntz (2007), p. 7.

[5] Depending upon the agent's coding system, storing a given feature in working memory can take up more or less space. As a result, the coding system affects what the agent is able to see.

[6] The types of codewords that are likely to be short are those that are either decision-relevant themselves or help agents reach decision-relevant conclusions.

for decision-making (Bordalo et al. (2023, 2012, 2013a,b, 2016b, 2020, 2024); Mullainathan et al. (2008)). One of their central ideas is that people *categorize* situations, grouping similar cases together and applying the same model of reasoning within categories. In doing so, they may transfer information from a situation where it is useful to one where it is not—which can explain the presence of framing effects. They also argue that agents tend to notice features that are prominent, contrasting, or surprising when comparing situations within the same category (for instance, an item that is particularly expensive compared to similar items the decision-maker recalls). A second key idea is that decision-makers only attend to some of the many features of a decision problem. Thus, the aspects of the problem that are salient play a critical role in determining what people choose.

Another approach to bounded rationality has been to assume that a decision-maker optimizes subject to some constraint. Constraints that have been considered include imperfect memory (Mullainathan (2002) and Wilson (2014)), limited information (Sims (2003)), limited attention to decision-relevant variables (Gabaix (2014)), limited precision in communication (Cremer et al. (2007)), limited ability to reason about other agents' strategies (Stahl and Wilson (1994), Crawford and Iriberri (2007)), cognitive uncertainty (Enke and Graeber (2023)), or bounded communication (Ellison and Holden (2014)).

Relative to the existing literature on bounded rationality, our contribution is to offer a theory of a central aspect of the human reasoning process: how one goes from seeing the parts to seeing the whole. Our theory highlights, in particular, the role that limited working memory plays in the reasoning process. Limited working memory not only constrains what the agent is ultimately able to see; it also leads to reasoning in steps, where—even when the agent receives no new information—insights unfold successively, as each new insight opens the door to others.[7]

Limited attention plays a central role in our framework, as in the work of BGS et al. Their work holds fixed what agents know and examines how attention to different

[7] Our model distinguishes between two types of memory: permanent storage (i.e. "hard drive"), which we assume to be infinite, and working memory (i.e. "RAM"). By contrast, there is only a single type of memory in the models of Mullainathan (2002) and Wilson (2014).

pieces of information affects decision-making. Our focus, by contrast, is on how limited working memory shapes the reasoning/learning process—and the consequences for what decision-makers ultimately know, or think they know.

Our paper also relates to a literature on narratives and mental models (e.g. Bénabou et al. (2018); Eliaz and Spiegler (2020); Gibbons et al. (2021); Schwartzstein and Sunderam (2021)).[8] Perhaps most closely related is Wojtowicz (2024), who considers a different mechanism that can lead to path dependence: the speed at which new data arrives.[9] Relative to the existing literature, we adopt a distinct definition: we think of narratives as understandings of the big picture.

Finally, our paper relates to the cognitive psychology literature on inductive reasoning, which examines how people infer theories from a limited set of examples. One prominent strand—the minimum description length (MDL) framework—argues, as we do, that people favor explanations that are encoded simply (see Chater (1996), Feldman (2000), and Chater and Vitányi (2003)). While MDL treats simplicity as a model-selection criterion, simplicity emerges endogenously in our model from a working-memory constraint.[10] Another contrast is that MDL focuses on Kolmogorov complexity/simplicity—defined as the length of the shortest computer program that encodes an explanation—whereas we allow for heterogeneity in what individuals consider simple, which captures the diversity in what people are able to see in the same data. A second influential perspective casts induction as Bayesian inference guided by fixed, hard-wired priors—often taking the form of hierarchical Bayesian networks (see Tenenbaum et al. (2006) and Oaksford and Chater (2007)). By contrast, in our framework, the agent needs to learn what model

[8] Schwartzstein and Sunderam (2021) conceive of an agent as choosing a mental model from an available set that is potentially influenced by a persuader. Eliaz and Spiegler (2020) conceive of narratives as causal interpretations of events (specifically, directed acyclic graphs). Gibbons et al. (2021) conceive of narratives in terms of categorizations and examine how a group's shared narrative affects their capacity to cooperate. Bénabou et al. (2018) model the process by which narratives disseminate.

[9] In Wojtowicz (2024), an agent repeatedly evaluates their existing model against "nearby" alternatives, and switches whenever the alternative has better fit. If data arrives too quickly, path dependence may arise and the agent may not converge to the maximum-likelihood model.

[10] That is, simple explanations are favored not because agents actively seek them out, but because agents struggle to develop complex explanations.

they are in without priors to guide them. Finally, whereas most work in psychology (like economics) emphasizes the outputs of inductive processes—the conclusions people ultimately draw—we examine how people build up partial insights step by step.

The paper proceeds as follows. Section 2 presents a version of the model in which the agent is purely deductive. Section 3 considers the possibility that the agent extrapolates as well. Section 4 shows how the model can be applied to the phenomenon of perceptual rivalry. Section 5 discusses the extension to a choice setting. Section 6 considers implications for the persuasion problem. Section 7 discusses other potential extensions. Section 8 concludes. All proofs are contained in the appendix.

# 2 Deduction

## 2.1 Model

An agent is presented with a picture $P$ of size $M \times N$. A picture is a matrix whose elements $p_{xy}$ take the values "black" or "white."

$$\begin{bmatrix} p_{11} & p_{12} & p_{13} & \cdots & p_{1N} \\ p_{21} & p_{22} & p_{23} & \cdots & p_{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{M1} & p_{M2} & p_{M3} & \cdots & p_{MN} \end{bmatrix}$$

We refer to the elements $p_{xy}$ as pixels.

The agent's task—described formally below—is to determine features of the picture. For example, the agent might be shown the picture in Figure 2 and their task might be to determine whether this picture depicts a smiley-face.

Let $R$ denote an $m \times n$ region of the overall $M \times N$ grid; and let $P_R$ denote the submatrix of $P$ on region $R$ (see Figure 3). Let $q$ be a proper, non-empty subset of the set of

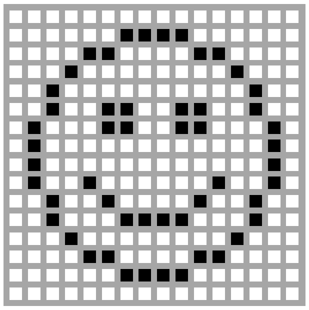


Figure 2: Example of a Picture

all pictures of size $m \times n$; we refer to $q$ as an $m \times n$-pattern. We say that pattern $q$ applies to $P_R$ if $P_R \in q$. For example, Figure 4(a) shows a set of pictures which we might refer to as the "smile" pattern. This pattern applies to region $R$ of the smiley-face picture (Figure 4(b)). Similarly, there might be an $M \times N$-pattern "smiley-face"; and we might say that $P$ depicts a smiley face if this pattern applies to $P$.[11]

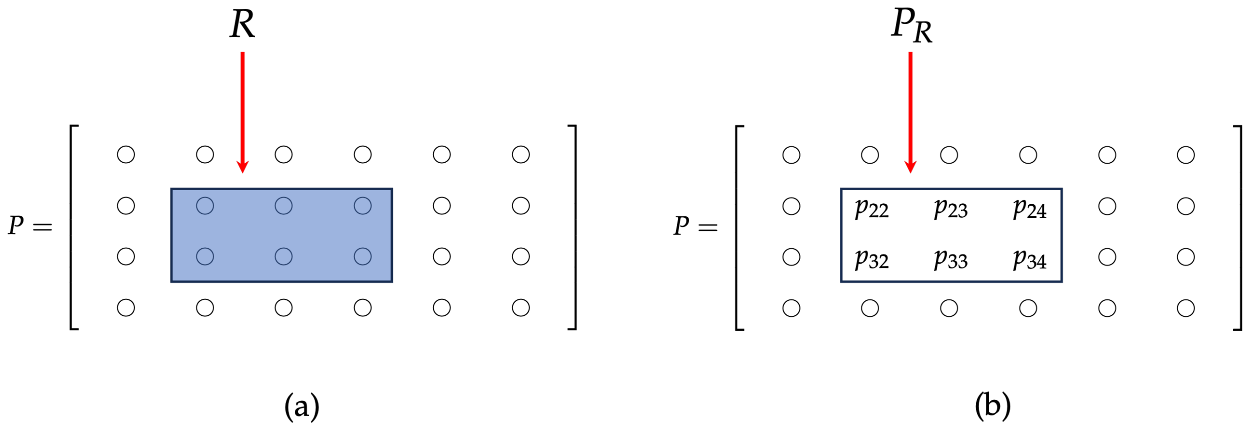
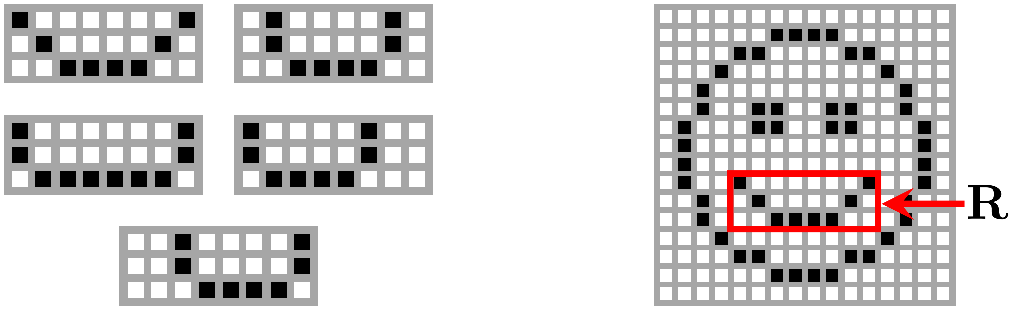


Figure 3

We refer to a pattern-region pair $f = (q, R)$ as a feature, and we say that feature $f$ applies to picture $P$ if pattern $q$ applies to $P_R$. Note that two features $(q, R)$ and $(q', R')$ might be *equivalent* in the sense that they imply the same restrictions on the picture:

$$\{P : P_R \in q\} = \{P : P_{R'} \in q'\}.$$

[11] Here, we define a "smile" in binary terms: a picture, or region of a picture, either depicts a smile or not (depending upon whether it belongs to some set). In Section 7.3, we introduce an alternative, non-binary way of categorizing where a picture (or region) may be more or less smile-like.



(a) The smile pattern

(b) The smile pattern in region $R$

Figure 4

We may sometimes refer interchangeably to equivalent features. For instance, we might say that a feature of the picture is "pattern $q$ on region $R$," even though this statement, strictly speaking, corresponds to an equivalence class of features.

The agent has a *knowledge set* consisting of features of $P$. The agent may add to this knowledge set over multiple periods, $t \in \{0, 1, 2, \ldots\}$. We denote the agent's time-$t$ knowledge set by $K_t$. If $f \in K_t$, we say that "at time $t$, the agent thinks feature $f$ applies to $P$."

The agent's initial knowledge set consists of knowledge of each pixel's color. That is, the agent knows the pattern that applies to every $1 \times 1$ region of the picture (whether it is "black" or "white") but nothing else: $K_0 = K_{\text{pixels}}$, where

$$K_{\text{pixels}} = \{(q, R) : q \text{ applies to } P_R,\, q \text{ is a } 1 \times 1\text{-pattern, and } R \text{ is a } 1 \times 1\text{-region}\}.$$

While the agent knows the color of every pixel—and thus possesses complete information about $P$—they lack any deeper understanding of what $P$ depicts—for example, whether it depicts a smiley-face. In order to have a deeper understanding, the agent would need to know features of larger regions of the picture.

The agent adds to their knowledge set by loading existing knowledge into *working memory* and drawing conclusions. Formally, let $W_t \subseteq K_t$ denote the knowledge the agent loads into working memory at time $t$. There is a set of features $\Delta_t$ that can be deduced purely from the knowledge in working memory. At time $t+1$, the agent adds these deductions to their knowledge set: $K_{t+1} = K_t \cup \Delta_t$.[12]

To give an example, suppose the agent loads the following features of the picture into working memory: "the number of white pixels in region $R$ is even" and "the number of white pixels in region $R$ is prime." Then, the agent concludes that "there are two white pixels in region $R$" and this is added to knowledge.[13]

Importantly, the agent has limited working memory, which restricts the amount of knowledge they can load into working memory and bounds their ability to make deductions. What the agent can load into working memory depends upon how they *encode information*.

The agent has a code $C : Q \rightarrow B$ that uniquely maps the set $Q$ of patterns (of any size) to the set $B$ of finite-length binary strings. For example, the agent might use "1101" to represent the "smile" pattern. We refer to "1101" as the agent's *codeword* for "smile." We assume that the agent's code is exogenously given.[14]

The amount of working-memory space a feature $f = (q, R)$ takes up depends upon how efficiently its pattern $q$ is encoded. A pattern $q$ takes up more space in working memory when it has a long codeword (e.g. "1001011") as opposed to a short codeword (e.g. "0"). Specifically, we assume that the agent's working memory constraint is:

$$\sum_{(q,R)\in W_t} \text{length}(C(q)) \leq L,$$

[12] Formally, we define $\Delta_t$ as follows. For a set of features $F$, let $\mathcal{P}^F \equiv \{P : P \text{ has all features } f \in F\}$. Then, $\Delta_t \equiv \{f : \mathcal{P}^{W_t} \subseteq \mathcal{P}^{\{f\}}\} - K_t$.

[13] To be precise, if the agent loads any feature in the equivalence class "the number of white pixels in region $R$ is even"—and likewise any feature in the equivalence class "the number of white pixels in region $R$ is prime"—they conclude (and add to knowledge) all features in the equivalence class "there are two white pixels in region $R$." Note that there are other (equivalence classes of) features that the agent deduces as well—such as "the number of white pixels is a power of two."

[14] Notice that the agent's encoding scheme assigns different codewords to patterns of different sizes, even when those patterns describe similar objects (for example, squares of different sizes). The agent might, alternatively, use a single default codeword for the "square" pattern and modify it with prefix codewords that represent transformations such as scaling or rotation. This might economize on the number and length of codewords. We rule out such encoding schemes purely for expositional ease.

where $C(q)$ denotes pattern $q$'s codeword, length$(C(q))$ denotes the length of codeword $C(q)$, and $L \geq 1$ denotes the agent's working memory capacity. Note that, in assuming that the amount of space feature $(q, R)$ takes up in memory depends only upon its pattern, we implicitly assume that regional information is encoded so efficiently that it takes up negligible space. Our results do not depend materially upon this assumption.

Given that patterns with short codewords take up relatively little space in working memory, we refer to these as "simple" patterns—in contrast to patterns with long codewords, which we refer to as "complex." We will also refer to features as simple when their associated patterns are simple. Observe that whether patterns/features are simple or complex depends upon how the agent encodes them. Thus, what is simple for one agent may be complex for another.

There are just two patterns for pixels (i.e. $1 \times 1$ regions): "white" and "black." We assume, for expositional ease, that the agent assigns one of these patterns codeword "0" and the other codeword "1" (i.e. the two binary strings of length one).[15] This assumption allows us to think of $L$ as the number of pixels the agent can load into working memory.

## 2.2 The Agent's Problem

For now, our focus will be on *what* the agent is able to deduce rather than on *how* the agent employs their deductions. However, we have in mind that, in the background, the agent faces a decision problem for which their deductions are relevant. We explicitly consider such a decision problem in Section 5.

In the remainder of this section, we examine what deductions the agent is capable of making given their code $C$. In other words, our approach will be to ask: is there a thought process (i.e. a sequence of working memories $W_0, W_1, ...$) that would allow the agent to reach the conclusion that some feature $f$ applies to picture $P$?[16]

[15] This assumption maximally favors the use of pixels in deduction by minimizing their working memory footprint.

[16] We do not take a stance on how much (calendar) time it takes the agent to complete a deduction step. We might think of the amount of time involved as the agent's "computational ability." For an agent with high computational ability, it is probably sufficient to examine—as we do—whether there is some deductive

We are agnostic for now about how exactly deductions translate to choice, but we have in mind that an agent makes better decisions when their deductions improve. This holds, for instance, in Section 5, where our agent makes a choice only after they reach a conclusion about the optimality of that choice.

## 2.3 What can be deduced?

Let us consider what the agent is capable of deducing—as well as how long deductions take. We start with the following definition.

**Definition 1.**

1. *A feature $f$ that applies to $P$ is deducible in $\tau$ steps ($f \in D_\tau$) if there exists a sequence $W_0, ..., W_{\tau-1}$ of working memories such that $f \in K_\tau$.*
2. *A feature $f$ that applies to $P$ is deducible ($f \in D$) if it is deducible in $\tau$ steps for some finite $\tau$.*

Let us also define what it means for the agent to know "everything about $P$" (i.e. have a complete understanding of the picture's features).

**Definition 2.** *We say that the agent knows "everything about $P$" if, for any feature $f$ that applies to $P$, the agent knows an equivalent feature.*

We obtain the following proposition.

**Proposition 1.**

1. *If $L \geq M \times N$: everything about $P$ is deducible in a single step.*
2. *If $L = 1$: the agent cannot deduce anything about $P$ (i.e. every feature in $D$ is equivalent to some feature in $K_0$).*

process ($W_1, W_2, ...$) by which the agent can reach a conclusion. For an agent with low computational ability, however, we might also wish to examine the number of steps involved. We make the modeling choice in this paper to focus primarily on the role of *limited working memory* rather than *limited computational ability*. However, it is easy to extend our analysis to consider limited computational ability as well.

3. *If $1 < L < M \times N$, for some picture $P$ and encoding $C$:*

   - *The agent can deduce something about $P$, but less than everything.*
   - *There are features that apply to $P$ that are only deducible in multiple steps.*

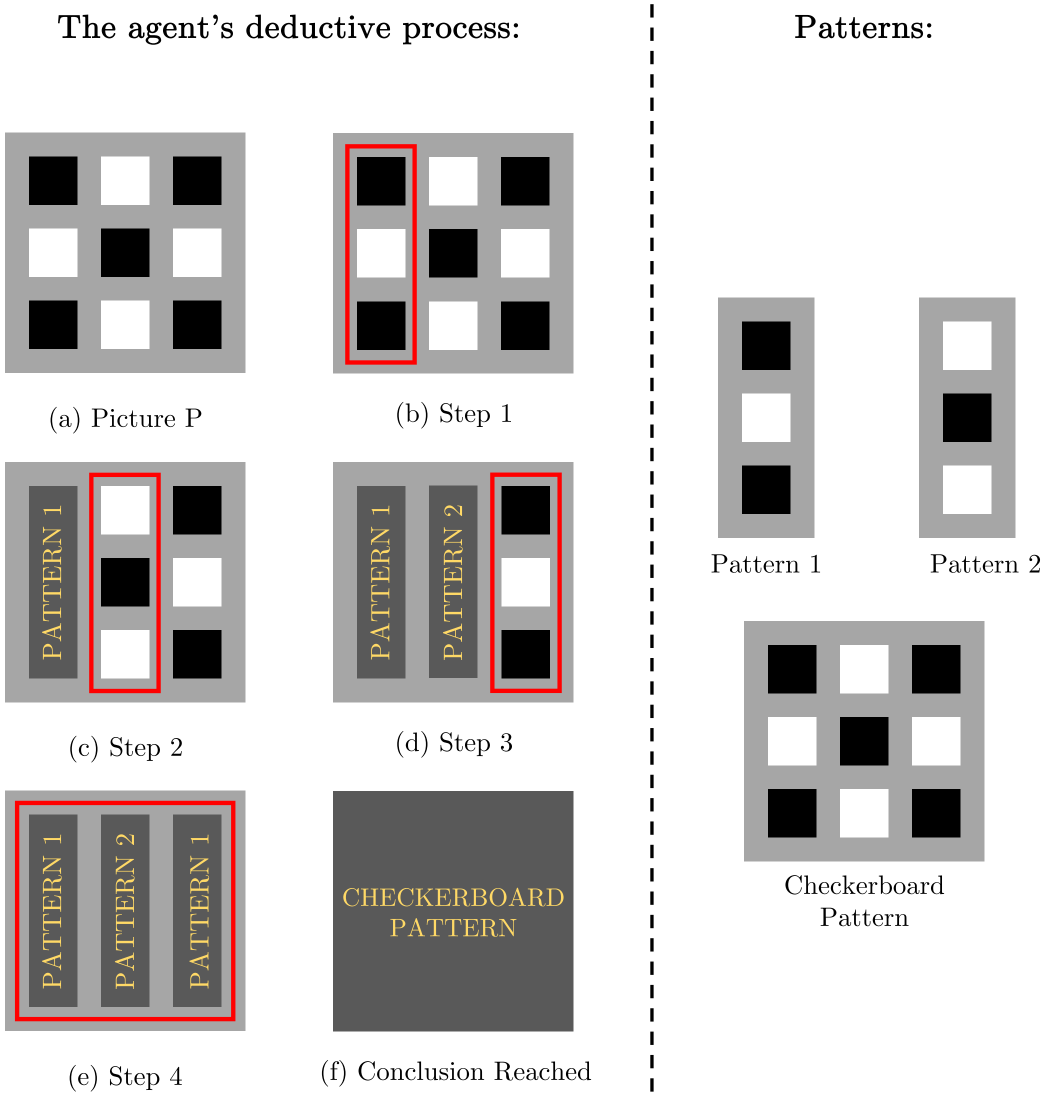
Part 1 of the proposition says that if the agent has sufficient working memory ($L \geq M \times N$), everything is deducible in a single step. Intuitively, if $L \geq M \times N$, the agent can load their knowledge of every pixel's color into working memory. If the agent has every pixel loaded in memory, they can deduce every feature of the picture. Therefore, only when $L < M \times N$ does working memory bound the agent's ability to make deductions.

Part 2 of the proposition says that if the agent has just one unit of working memory ($L = 1$), the agent cannot deduce anything. The reason is that the agent learns the big picture by *combining* or *integrating* their knowledge of features. If $L = 1$, they can only load a single feature into working memory.

To understand Part 3, consider Figure 5 and suppose the agent has working memory capacity of six. While the "checkerboard pattern" applies to picture $P$, the agent cannot work this out in a single step since they cannot load all nine of $P$'s pixels into working memory. However, they can work it out in multiple steps if they have a suitable code.

Consider, for instance, a code where patterns 1 and 2 (depicted in Figure 5) each have codewords of length two. The agent can load the three pixels on the left side of $P$ in working memory and deduce that pattern 1 applies to the left column (panel b). In subsequent steps, the agent learns that patterns 2 and 1 apply to the middle column and right column respectively (panels c and d). Using all six units of working memory, the agent can then load their knowledge of each column's pattern and deduce that $P$ is a checkerboard (panel e).[17]

[17] In fact, if "pattern 3"—consisting of the six pixels in the middle and right columns—also has a codeword of length two, then a memory capacity of four ($L = 4$) is sufficient to work out that $P$ is a checkerboard. Using three units of memory, the agent can learn the pattern of each column. Using four units of memory, the agent can then load their knowledge of the middle- and right-column's patterns and learn that pattern 3 applies. Finally, using four units of memory, they can learn that $P$ is a checkerboard by storing their knowledge that pattern 1 applies to the left column and pattern 3 applies to the rest of the picture.


The agent's deductive process:

Patterns:

(a) Picture P

(b) Step 1

(c) Step 2

(d) Step 3

(e) Step 4

(f) Conclusion Reached

Pattern 1

Pattern 2

Checkerboard Pattern

Figure 5: A Multi-step Deduction

We can also use the checkerboard example to show that the agent may be able to deduce something but not everything. For example, suppose that the agent has a code where all patterns—except the patterns for pixels—have codewords of length $L+1$ or greater. Then, the only patterns the agent can load into working memory are those for pixels. It is thus impossible for the agent to learn that $P$ depicts a checkerboard. However, the agent can still learn something. For example, by storing the three pixels on the left in working memory, they can learn that pattern 1 applies to the left column.

## 2.4 Chunking

If we examine the multi-step deduction depicted in Figure 5, the key thing the agent does between steps is condense information. Specifically, the agent combines multiple features, with a high memory demand, into fewer features with a lower memory demand. For instance, the three pixels in the left column of $P$ are initially represented as three features—one feature for each pixel—which requires three units of working memory. These features are then combined into a single "pattern 1" feature, which requires only two units of working memory.[18]

We refer to this process of combining features as "chunking" in light of a closely related literature in cognitive psychology. The term "chunk" was introduced by Miller (1956), who made a remarkable discovery about short-term memory. Miller found that the ease or difficulty of remembering information depends upon how it is stored. For example, a telephone number can be stored as a string of individual digits (e.g. 5, 1, 2, 3, 4, 8, 2) or as a set of larger numbers, or chunks (e.g. 512, 3482). Chunking the phone number, he discovered, significantly reduces the strain on memory.

William Chase and Herbert Simon further developed the concept of chunking in a famous 1973 paper.[19] They conducted an experiment in which expert and novice chess players were shown board positions for five seconds and then asked to reconstruct the positions from memory. When participants were presented positions from *actual* chess games, the experts significantly outperformed the novices. On average, experts correctly recalled the locations of 16 chess pieces, while novices only correctly recalled the locations

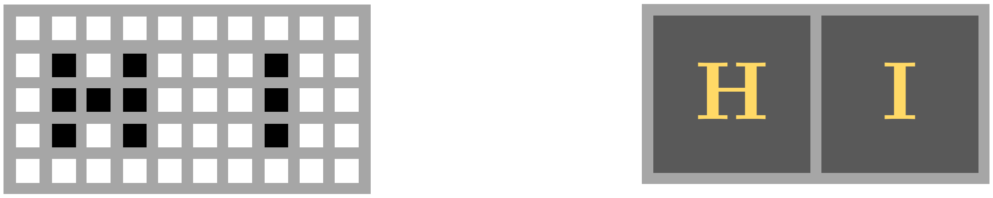
[18] Chunking relates to the literature on data compression—in particular work on minimum description length (MDL) (see Rissanen (1978) and, for an overview, Grünwald (2007)). The MDL literature points out that "total description length" is equal to the sum of "description length of the model $M$" and "description length of the data $D$ given the model $M$." Consequently, there is a tradeoff between using a complex model, where data description is simple, and a simple model, where data description is complex. This tradeoff is present in our framework. For instance, the left column in Figure 5 could be represented as "pattern 1" or as three pixels. The first representation economizes on data description (it only involves a single feature, compared with three features in the second representation) whereas the second representation economizes on model description (it is composed of pixels, which are simpler features than "pattern 1").

[19] Chase and Simon (1973) build on earlier experimental results of De Groot (1965).

of 4 pieces. However, when participants were shown *random* board positions, there was no difference between experts and novices; furthermore, both groups performed even worse than the novices did on actual board positions.

Chase and Simon posited that expert players have a greater ability to chunk typical chess configurations. In terms of the model, we can think of experts and novices as having different encodings of patterns. Experts assign short codewords to common board positions, whereas novices do not. This allows experts to represent a board position in terms of just a few simple features, in contrast to novices.

To make this point more concrete, consider Figure 6. Panel (a) shows a constellation of pixels that we might think of as akin to the position of pieces in a board game. Panel (b) shows that one way of representing this configuration is as an "H" and an "I." Suppose one agent (akin to the expert chess player) has short codewords for the "H" and "I" patterns—as they might if they are familiar with the Roman alphabet. Say both codewords have length two. If this agent stores the board in working memory as an "H" and "I", it takes up just four units of memory—far less memory than the fifty units needed to store fifty individual pixels. Suppose a second agent (akin to the novice) has long codewords for "H" and "I." This agent may not find it any better to store the board as an "H" and an "I" than as fifty pixels.



(a) Game Board

(b) Game Board - Chunked

Figure 6: Chunking a Game Board

## 2.5 Bottom-up and Top-down Thinking

Cognitive psychologists make a distinction between bottom-up and top-down thinking (see Marr (2010); Neisser (2014)). Our model captures this distinction. We refer to the agent's thinking as bottom-up if they make deductions about larger regions from features of smaller regions. For instance, deducing that a set of pixels form an "H" (see Figure 6) is bottom-up. In fact, chunking more generally is bottom-up. Notice that when the agent engages in bottom-up thinking, they are *constructing narratives* (i.e. big-picture statements).[20]

Top-down thinking, by contrast, involves making deductions about smaller regions from features of larger regions. For instance, if the agent uses their knowledge that $P$ depicts a smiley-face to deduce that region $R$ depicts a smile (see Figure 4), this would be top-down. When the agent engages in top-down thinking, they are *using narratives*, rather than constructing them.

The following is a formal definition of these two types of thinking.

**Definition 3.** *Suppose the agent deduces feature $f = (q, R)$ at time $t$.*

- *This deduction involves "bottom-up thinking" if, for all features $g = (q', R') \in W_t$, $R'$ is strictly contained in region $R$.*
- *This deduction involves "top-down thinking" if there is a feature $g = (q', R') \in W_t$ with region $R$ strictly contained in $R'$, and the agent does not learn $f$ if $g$ is omitted from $W_t$.*

Unchunking information—for instance, taking an "H" and deducing the constituent pixels—fits our definition of top-down thinking. Our definition of top-down thinking also allows the agent to make deductions using a combination of big-picture and smaller-picture features. For example, the agent might use their knowledge that picture $P$ is a smiley-face (i.e. belongs to a set of smiley-face pictures) in combination with their knowledge of a few pixels to deduce that region $R$ is a "smile." Note that not all types of

[20] An issue with defining a feature $f = (q, R)$ as a narrative based on whether region $R$ is large is that there might be an equivalent feature $f' = (q', R')$ where $R'$ is small. There are various alternative definitions of narratives that deal with this issue. For instance, take all of the features in $f$'s equivalence class and intersect their associated regions. Let $R^*(f)$ denote this intersection—which is, in fact, the region of some feature in $f$'s equivalence class. We might refer to $f$ as a narrative if the region $R^*(f)$ is large. Another alternative would be to refer to $f$ as a narrative if it "involves" a large number of pixels, where a pixel is involved if $f$ is informative about its color (more precisely, there is some information about the remaining pixels' colors that, in combination with $f$, would be sufficient to determine the pixel's color). For simplicity, in the remainder of the paper, we will refer to $f = (q, R)$ as a narrative if region $R$ is large; but our insights carry over to these alternative definitions.

thinking can be classified as bottom-up or top-down. The agent would be using a hybrid form of thinking, for instance, if they were to deduce a feature $f$ of region $R$ from two other features of region $R$, $g$ and $h$.

The following proposition shows that there are features that can only be deduced using bottom-up thinking *and* there are features that can only be deduced using top-down thinking.

**Proposition 2.** *There exists picture $P$, code $C$, parameter $L$, and deducible features $f$ and $f'$ such that:*

*(a) In any working memory sequence where $f$ is deduced, the deduction of $f$ involves bottom-up thinking.*

*(b) In any working memory sequence where $f'$ is deduced, the deduction of $f'$ involves top-down thinking.*

Given that the agent starts with information that is pixellated, their basic task is to move from a small-picture understanding to a bigger-picture understanding. Thus, it is not surprising that some deductions require bottom-up thinking.

It is less obvious that there are deductions requiring top-down thinking. Starting from pixels, top-down thinking seems like a circuitous route to deducing a feature $f$. However, the bottom-up path might be blocked if there are no simple intermediate features the agent can use to chunk up to $f$. Even if the direct path is blocked, there might be simple intermediate features that allow the agent to chunk up to a simple, big-picture feature $f'$ that is informative about $f$. In such a case, the agent might take a detour by first deducing $f'$, then engaging in top-down thinking to deduce $f$ from $f'$ (see Figure 7 for further illustration). To summarize, a key reason why the agent uses top-down thinking is that narratives provide a simple way of thinking about the picture that is not memory-intensive.

Cognitive psychologists have conducted a variety of experiments that suggest the use of top-down thinking (see Gilbert and Li (2013)). For example, Gregory (1970) discusses

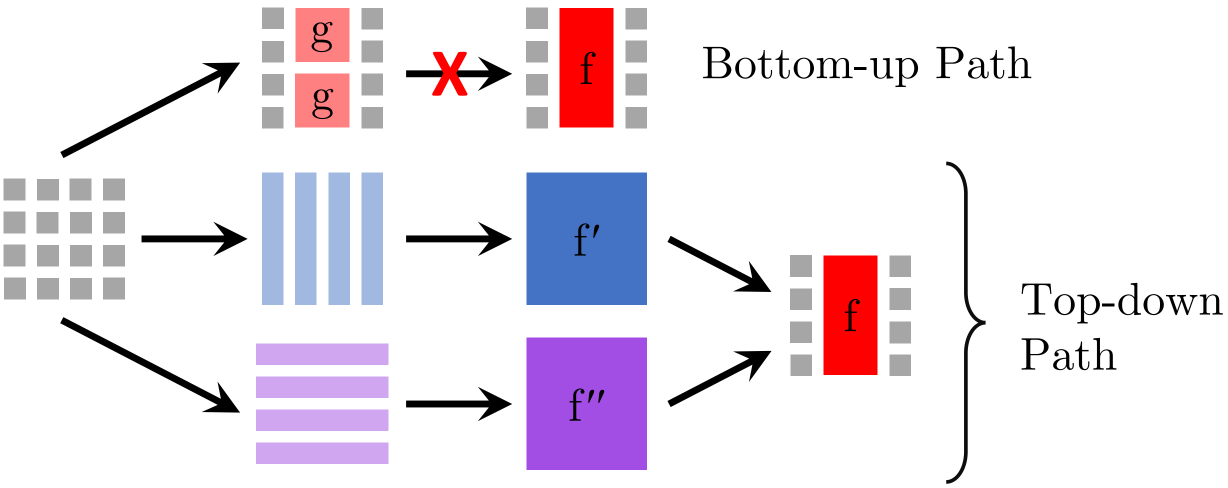


Figure 7: This picture illustrates why top-down thinking may be necessary. First, it shows that a bottom-up path to deducing feature $f$ may be blocked because the intermediate features ($g$) are complex. Second, it shows that there may be simple big-picture features ($f'$ and $f''$) that can nonetheless be used to deduce $f$.

the hollow mask illusion, where people perceive a hollow (concave) face mask as convex, like a normal face. Gregory argues that this illusion demonstrates the use of a pre-existing narrative—that the picture depicts a convex face—to interpret the visual information.

## 2.6 Optimal Codes

We opt to think of the agent's code $C$ as something they are endowed with, rather than a choice. Nonetheless, there is reason to think that it evolves. Chase and Simon (1973)'s experiment provides suggestive evidence. Their results could be purely the result of selection (i.e. better chess players are endowed with better codes); but it seems more likely that better chess players have better codes because they have more experience with chess.

This raises a variety of questions—such as how codes evolve. However, an even more basic question is what we mean by a "better" code or an "optimal" code. Here, we give one possible definition of optimality and discuss its properties.

In order to talk about optimality, we first need to define an objective. With this in mind, we define a task $\tau = (F, \mathcal{P})$ for the agent based on a particular set of features ($F$)

and a particular set of pictures ($\mathcal{P}$). We say that the task is *achievable* if, for each picture $P \in \mathcal{P}$, the agent is able to deduce all features $f \in F$ that apply to $P$.

For a given task and code, let $L^{\min}(\tau, C)$ denote the minimum amount of working memory capacity needed for the task to be achievable. We say that a code is $\tau$-optimal if it minimizes the amount of working memory needed for the agent's task $\tau$.

**Definition 4.** *Let $C^*(\tau) = \arg\min_C \{L^{\min}(\tau, C)\}$ denote the code (or set of codes) that minimize the working memory needed for task $\tau$. We will refer to $C^*(\tau)$ as the optimal code(s) for task $\tau$.*

It is relatively easy to show that the optimal code depends upon the task, which we state as Proposition 3.

**Proposition 3.** *The optimal code is task-specific. That is, for some tasks $\tau$ and $\tau'$, $C^*(\tau) \cap C^*(\tau') = \varnothing$.*

To gain intuition for this result, recall two of the tasks we have considered. In Figure 5, the agent's task is to identify whether a picture is a checkerboard. In Figure 6, the agent's task is to identify board positions composed of letters. In each case, working memory is conserved by assigning short codewords to particular patterns. In the first case, it makes sense to assign short codewords to patterns 1 and 2. In the second case, it is ideal to assign short codewords to the "H" and "I" patterns.

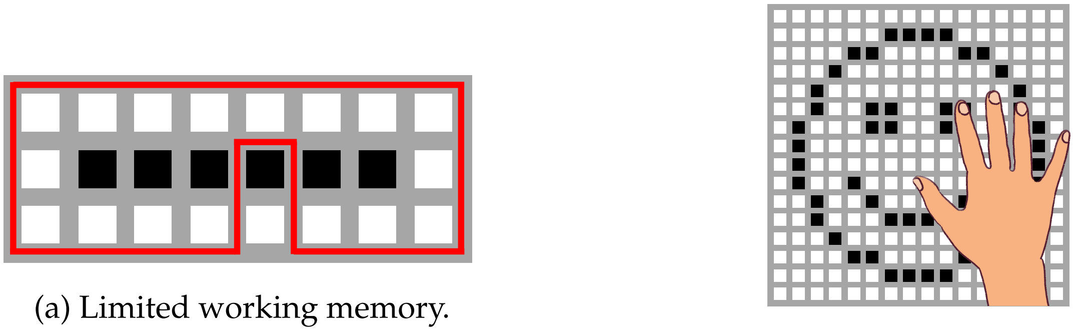
The number of short codewords is limited. Consequently, a code that is geared towards the first task (detecting checkerboards) is less geared towards the second task (detecting letters).

This result suggests that agents may have different competencies. One agent might have a code that leads them to excel at chess, while another might have a code that leads them to excel at Go. We will discuss later how the model might be profitably extended to environments where agents have different competencies.

# 3 Extrapolation

So far, we have assumed that the agent uses deduction alone to learn about the picture. While, in some cases, deduction may be sufficient to learn about the picture; in others, the agent may find it useful to employ an additional tool: extrapolation.

Figure 8 shows two cases where extrapolation could be useful. In panel (a), the agent's task is to detect whether the picture contains a horizontal line. The agent can load twenty-two pixels into working memory, but not the twenty-four pixels needed to deduce that the picture contains a line. To keep the example simple, suppose loading pixels is the only way the agent can detect the line. It might make sense in this case for the agent to extrapolate (i.e. decide that there is a line even though they cannot be sure).



(a) Limited working memory.

(b) Limited observation.

Figure 8: Benefits of Extrapolation

In panel (b), the agent is not able to see all of the pixels of the picture. Formally, we can think of this as a case where the agent's initial knowledge set $K_0$ contains only some of the pixels of $P$. Even if the agent has unlimited working memory, they cannot deduce that the picture depicts a smiley face; nonetheless, this would be a reasonable conclusion to draw. Thus, here too, extrapolation may have value.

We now consider a version of the model where the agent makes extrapolations, adding features to the knowledge set whenever they "fit" sufficiently well with the facts in working memory.

## 3.1 Model

As in the deduction model, we assume that the agent's initial knowledge set consists of pixels of $P$. However, we now generalize by assuming that the agent may initially know only a subset of $P$'s pixels: $K_0 \subseteq K_{\text{pixels}}$. This captures the case described in Figure 8(b).

We again assume that the agent loads a subset of their knowledge into working memory at time $t$ ($W_t \subseteq K_t$). However, we now assume that, at time $t$, the agent weighs whether to add a particular feature $f \notin K_t$ to the knowledge set, and this feature takes up space in working memory. Thus, the agent's working memory constraint is:

$$\sum_{(q,R) \in W_t \cup \{f\}} \text{length}(C(q)) \leq L.$$

The agent adds feature $f$ to the knowledge set if its fit with the "facts" in working memory ($W_t$) weakly exceeds a threshold $\alpha$ (with $0 < \alpha \leq 1$). A higher threshold $\alpha$ corresponds to an agent who is more deductive. We denote feature $f$'s fit with the facts by $\Phi(f, W_t) \in [0, 1]$.[21]

We assume a particular functional form for the fit function, with the intuitive property that fit is high when most pictures with features $W_t$ also have feature $f$.[22] To formalize the definition of fit, let $Z(S, R)$ denote the number of plausible pictures for region $R$ given a set of features $S$. An $m \times n$-size picture $B$ is plausible for region $R$ given $S$ if there is an $M \times N$-size picture $P'$ with features $S$ that corresponds to $B$ on region $R$ ($P'_R = B$). We define the fit of feature $f = (q, R)$ with $W_t$ as follows:

$$\Phi(f, W_t) = \frac{\log(Z(\varnothing, R)) - \log(Z(W_t, R))}{\log(Z(\varnothing, R)) - \log(Z(W_t \cup \{f\}, R))}, \tag{*}$$

where $\varnothing$ denotes an empty set of features. Notice that, in equation (*), the fit is the same regardless of the choice of logarithmic base. In the special case where no picture is consis-

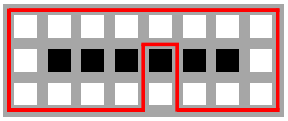
[21] Formally, $K_{t+1} = K_t \cup \{f\}$ if $\Phi(f, W_t) \geq \alpha$.

[22] The appeal of this fit function is its simplicity, but there may be other fit functions worth considering.

tent with the "facts" ($Z(W_t, R) = 0$), the fit function is undefined.[23] In such circumstances, we assume that $f$ is not added to the knowledge set.[24]

The fit function has several appealing attributes. For instance, $\Phi(f, W_t) = 1$ if and only if every picture with features $W_t$ has feature $f$, in which case $f$ is deducible from $W_t$.[25] Likewise, $\Phi(f, W_t) = 0$ if and only if no picture with features $W_t$ has feature $f$. Fit is also equal to zero if there are no facts in working memory ($W_t = \emptyset$). Thus, the agent never makes extrapolations when they have no facts.

To get a better sense of how the fit function works, let us return to the horizontal-line example. Suppose the agent loads the highlighted pixels from Figure 9(a) into working memory and evaluates the horizontal-line feature ($f$) shown in Figure 9(b). Let $R$ denote the region that covers the entire picture. Observe that $Z(\emptyset, R) = 2^{24}$ (there are $2^{24}$ possible pictures when the agent has no information about the pixels' colors), $Z(W_t, R) = 2^2$ (there are $2^2$ pictures consistent with $W_t$ since two pixels are not pinned down), and $Z(W_t \cup \{f\}, R) = 1$ (there is just a single picture consistent with $f$ and $W_t$). Thus, the fit of the horizontal-line feature with the pixels in working memory is: $\Phi(f, W_t) = \frac{\log 2^{24} - \log 2^2}{\log 2^{24} - \log 2^0} = \frac{24-2}{24-0} = \frac{11}{12}$.


(a) Working memory ($W_t$).

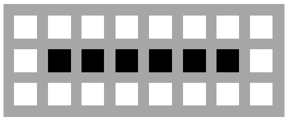

(b) Feature $f$.

Figure 9: Example - Fit Function

[23] If the agent is purely deductive, some picture must be consistent with the facts. However, when the agent extrapolates, some of their "facts" may be wrong; as a result, it is conceivable that no picture fits the facts.

[24] One may interpret $\log(Z(S, R))$ as a measure of the entropy on region $R$ given $S$—specifically, it corresponds to Boltzmann entropy. Under this interpretation, $f$ is a good fit with $W_t$ when adding it to $W_t$ has only a slight effect on entropy. Boltzmann entropy and Shannon entropy are closely related. They only differ by a scalar when all states of a system are equally probable. Thus, $\log(Z(S, R))$ could also be interpreted as region $R$'s Shannon entropy if we introduce a probability distribution over matrices on region $R$ where each plausible submatrix (given $S$) is equally likely to arise.

[25] Consequently, an agent with a fit threshold of one ($\alpha = 1$) is purely deductive.

## 3.2 Results

### 3.2.1 Misconceptions

A benefit of extrapolation is that it enables the agent to learn more of picture $P$'s features. In this sense, extrapolation reduces type I error. However, a cost of extrapolation is that the agent commits more type II errors, learning features that do not actually apply. We refer to these type II errors as "misconceptions."

To formalize this point, we define an analog of deducibility: we say that a feature $f$ is extrapolable if there is a sequence of working memories $(W_0, f_0), ..., (W_{\tau-1}, f_{\tau-1})$ such that $f \in K_\tau$. We denote the set of extrapolable features by $E$.

Let us focus for now on the case where the agent knows all of the pixels initially ($K_0 = K_{\text{pixels}}$). In this case, $E$ can be divided into a set $E^I$ of extrapolable features that apply to $P$ and a set $E^{II}$ of extrapolable features that do not apply. The following proposition shows that, the more the agent extrapolates (i.e. the lower the threshold $\alpha$), the larger are sets $E^I$ and $E^{II}$. In this sense, extrapolation decreases type I error but increases type II error.[26,27]

**Proposition 4a.** *Suppose the agent initially knows every pixel of $P$ ($K_0 = K_{pixel}$).*

1. *The sets $E^I$ and $E^{II}$ are both weakly decreasing in $\alpha$.*
2. *When $\alpha = 1$, $E^{II} = \emptyset$.*

**Patchwork Quilts**

The features in the agent's knowledge set might be internally consistent or internally inconsistent. We refer to an inconsistent knowledge set as a "patchwork quilt." To illustrate, suppose the agent's knowledge set contains the following features: "the number of black pixels in $P$ is prime," "the number of black pixels in $P$ is even," and "there are more

[26] The tradeoff between type I and type II error, first discussed by Neyman and Pearson, has been examined by many economists—for instance Sah and Stiglitz (1986), who argue that hierarchies tend to generate more type I error while polyarchies tend to generate more type II error.

[27] One may think of extrapolation as a form of lossy compression: by extrapolating, the agent economizes on the amount of working memory needed to reach a conclusion, but introduces the possibility of errors.

than two black pixels in $P$." This is a patchwork quilt since no picture possesses all of these features.

If the agent's knowledge set is a patchwork quilt, the agent definitely has misconceptions. Of course, even if the agent's beliefs are internally consistent, they may still have misconceptions. In this sense, a patchwork quilt is a special type of misconception.

It is easy to show that patchwork quilts can arise. This contrasts with the standard Bayesian model in which an agent's beliefs must be internally consistent. We will return to the topic of patchwork quilts when we consider a version of the model in which the agent can revisit and revise their past extrapolations.

### 3.2.2 Prediction

Having discussed the case where the agent observes all of the pixels ($K_0 = K_{\text{pixel}}$), let us now discuss the case where they only observe a subset ($K_0 \subset K_{\text{pixel}}$). In this situation, there is an additional benefit of extrapolation: it allows the agent to *predict* unobserved pixels. For example, in Figure 8(b), extrapolation allows the agent to predict the pixels covered by the hand.

When $K_0 \subset K_{\text{pixel}}$, we can partition set $E$ into three subsets: features that apply to picture $P$ ($E^I$), features that do not apply ($E^{II}$), and features that may or may not apply ($E^{III}$).[28] For instance, in Figure 8(b), there are features that may or may not apply depending upon what is covered by the hand. We can think of $E^{III}$ as the agent's *predictions* as their truth value is not ascertainable. The following generalization of Proposition 4a shows that extrapolation increases the size of $E^{III}$; in this sense, it allows the agent to make predictions.

**Proposition 4b.**

1. *The sets $E^I$, $E^{II}$, and $E^{III}$ are weakly decreasing in $\alpha$.*

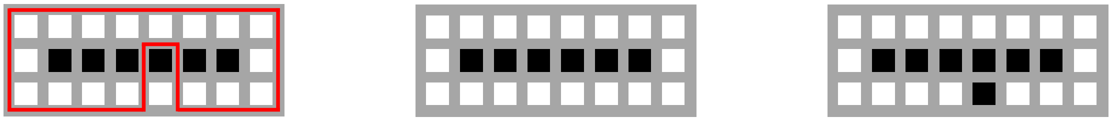
[28] Let $\mathcal{P}^{K_0} = \{P' : P' \text{ has all features } f' \in K_0\}$. Feature $f$ applies to picture $P$ if it applies to all of the pictures in $\mathcal{P}^{K_0}$; it does not apply to $P$ if it does not apply to any picture in $\mathcal{P}^{K_0}$; and it may or may not apply if it applies to some, but not all, of the pictures in $\mathcal{P}^{K_0}$ but not all.

2. *When $\alpha = 1$, $E^{II} = E^{III} = \emptyset$.*

### 3.2.3 Simplicity

In the extrapolation model, the agent is more likely to adopt features (or explanations) that are coded as simple.[29] This follows from the assumption that considering a feature for adoption consumes working memory.

To illustrate, let us revisit the horizontal-line example. Figure 10 shows two features, $f$ and $f'$, that the agent might consider adding to their knowledge set. Recall that feature $f$ has fit of $\frac{11}{12}$, and it is straightforward to show that feature $f'$ has the same fit. Given that they have the same fit, one might be tempted to think that they are equally likely to be adopted. However, if $f$ is a simple feature (i.e. "horizontal line" has a short codeword) while $f'$ is a complex feature, the agent might have enough memory capacity to adopt $f$ but not $f'$. In this way, the simple explanation ($f$) may be extrapolable while the complex one ($f'$) is not.



(a) Working memory ($W_t$).

(b) Feature $f$.

(c) Feature $f'$.

Figure 10: Example - Simplicity

The following proposition formalizes this point. It shows that features are more likely to be adopted when they are simpler.

**Proposition 5.** *Suppose $f = (q, R)$ is a feature with pattern $q$. Consider two codes $C$ and $C'$ that differ in only one respect: pattern $q$ has a shorter codeword in $C$ than $C'$. Let $E$ and $E'$ denote the set of extrapolable features under each code. If $f$ is in $E'$, then $f$ is also in $E$. Moreover, there exist instances of codes $C$ and $C'$ and parameter values such that $f \notin E'$ and $f \in E$.*

[29] Relatedly, Gestalt psychology's principle of *prägnanz* holds that perception gravitates towards simple, parsimonious perceptions, even in complex environments (e.g. Wertheimer (1938)).

One implication of this proposition is that the agent has a tendency to adopt and employ narratives that are relatively simple. Indeed, a simple, incorrect narrative might be adopted over a complex, correct one.

# 4 Perceptual Rivalry

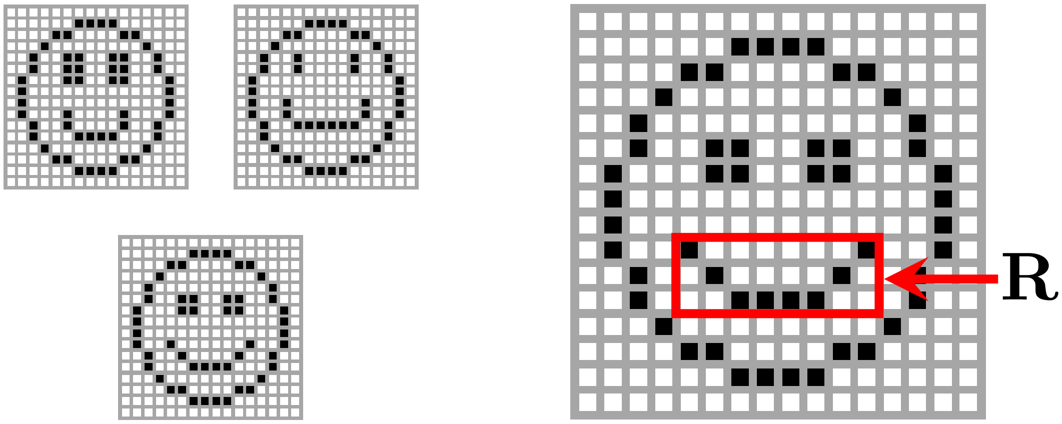
Consider the rabbit-duck illusion (Figure 1(b)). A key property of this illusion is that people have trouble seeing a rabbit once they have seen a duck—and vice-versa. It is possible to switch from one perception to the other, but it is hard. Moreover, it is virtually impossible to see rabbit and duck at the same time. This is an example of a more general phenomenon. Once someone has one way of perceiving a picture, it is hard to perceive the picture in other ways (see Leopold and Logothetis (1999)).

In line with a cognitive psychology literature on salience (see Serences and Yantis (2006)), we capture this idea by assuming that "powerful" narratives are salient; their salience can make it hard for competing narratives to emerge. For instance, once the agent has perceived "rabbit," the salience of "rabbit" may block "duck." This naturally leads to the possibility of multiple stable perceptions, where what the agent ultimately perceives (e.g. rabbit or duck) depends upon the agent's initial working memory sequence.[30]

## 4.1 Power

We assume that a feature's salience depends upon its "power"—a concept we will now define. To illustrate the concept of "power," consider Figure 11. Figure 11(a) shows a "smiley-face" feature, which consists of three full-picture matrices that all depict smiley-faces. If the agent has "smiley-face" in working memory, it helps them evaluate region $R$ of picture $P$ (see Figure 11(b)): in fact, in combination with knowledge of just one extra pixel, it becomes clear that the "smile" pattern of Figure 4(a) applies. For this reason, we think of "smiley-face" as a powerful feature for region $R$.

[30] In this case, there may be several knowledge sets the agent can reach—rather than a single extrapolable



(a) The "smiley-face" feature.

(b) The power of "smiley-face."

Figure 11: Example of Power

More generally, we say that a feature $f$ is powerful for region $R$ if it narrows down the possible subpictures on $R$. Formally, we define power as follows.

**Definition 5.** *The "power" of a set of features F for region R is:*

$$\Gamma_R(F) = 1 - \frac{\log Z(F, R)}{\log Z(\varnothing, R)}.$$

Notice that power takes the value zero when a set of features $F$ does not narrow down the possibility set at all. By contrast, power takes the value one when $F$ perfectly pins down what happens on region $R$ (since, in this case, $Z(F, R) = 1$).[31]

### Power and Working Memory

We make the assumption that features are more salient—that is, more likely to enter into working memory—when they are more powerful. Take the smiley-face feature in Figure 11, for instance. If "smiley face" is powerful, then once the agent recognizes that the picture depicts a smiley face, it becomes a highly salient feature.

We further assume that salience alone does not guarantee that a feature enters into

set $E$.

[31] One may think of the power of a set of features $F$ as the reduction in entropy on region $R$ relative to complete ignorance.

working memory—since there is also a question of whether including a feature strains the agent's memory capacity. However, we assume that features enter working memory provided they are sufficiently salient and sufficiently simple. Assumption 1, stated below, formalizes this idea.

**Assumption 1.** *Suppose, at time $t$, the agent is evaluating a feature $f = (q, R)$. Consider $f' \in K_t \setminus f$ and $F'' \subset K_t \setminus f$, where $\{f'\}$ is weakly more powerful than $F''$ on region $R$, $f'$ is weakly simpler than the sum of $f'' \in F''$, and at least one of these two inequalities is strict. If the agent loads every element of $F''$ into working memory ($W_t$), then they must also load $f'$.*

Notice that, under Assumption 1, the agent will substitute a set of features with an equivalent but simpler "chunked" feature (e.g., "H" in place of the pixels representing "H").[32] More generally, prioritizing simple, powerful features—as the agent does under Assumption 1—helps them economize on working memory.

## 4.2 Multiple Stable Perceptions

Here, we show that under Assumption 1, multiple stable perceptions may exist. For example, depending on their initial thoughts, the agent might perceive a rabbit (and no duck) in the long run, or a duck (and no rabbit). Proposition 6 formalizes.

**Proposition 6.** *Under Assumption 1, for some $(\alpha, L, P, C, K_0)$, there exist features $f$ and $f'$ and initial working memory sequences (i) $((W_0, f_0), ..., (W_t, f_t))$ and (ii) $((W'_0, f'_0), ..., (W'_t, f'_t))$ such that:*

- *Under initial sequence (i), $f \in K_\tau$ and $f' \notin K_\tau$ for all $\tau > t$ for any subsequent working memory sequence,*
- *Under initial sequence (ii), $f' \in K_\tau$ and $f \notin K_\tau$ for all $\tau > t$ for any subsequent working memory sequence.*

[32] Note that the agent could have the unchunked features in working memory—but only if they have the simpler, chunked feature as well.

To gain intuition for this result, let us continue with the rabbit-duck example. Suppose that "rabbit" and "duck" are both simple features and that there is a point in time where the agent is able to extrapolate to either "rabbit" or "duck." If the agent extrapolates to "rabbit," this may prevent the agent from later extrapolating to "duck" (and vice-versa). To see why, notice that once "rabbit" is in the knowledge set, it is likely to be *powerful*—and thus salient—in evaluating other features. Thus, if the agent considers adopting "duck," "rabbit" will be in working memory—and "duck" will be rejected since it has low fit with "rabbit" (see Figure 12(a)). We see then that, in the long-run, the agent may end up adopting "rabbit" or "duck"—but not both.

Notice that "rabbit" also tends to block the adoption of duck-related features—such as "duck bill" for the left-hand side—as they have low fit with "rabbit" (see Figure 12(b)). By contrast, features related to "rabbit"—such as "rabbit ears" for the left-hand side—are likely to be adopted (see Figure 12(c)). Thus, when the overall picture is perceived as a

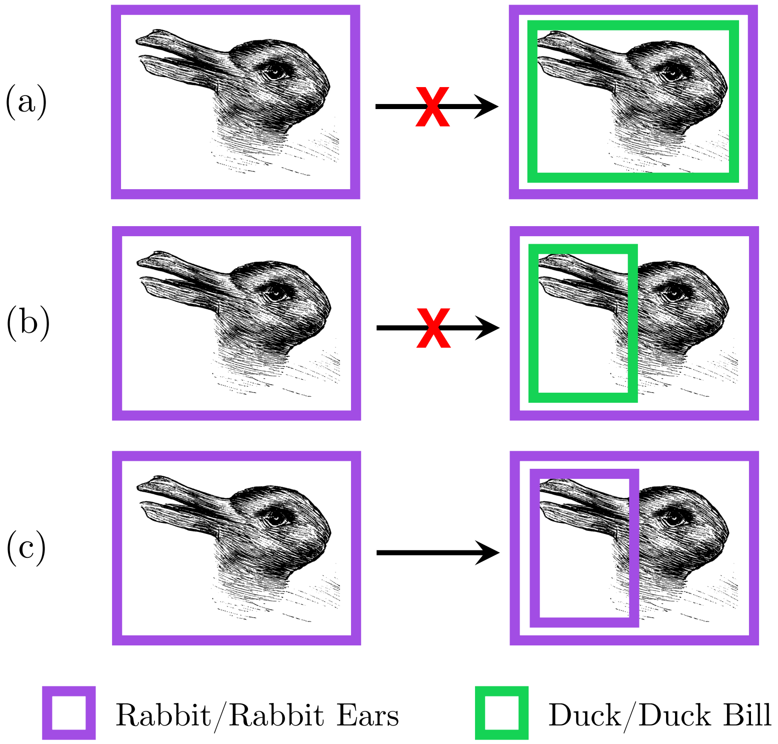


Figure 12: Multiple Stable Perceptions

"rabbit," the agent also sees the regions as rabbit-like.

## 4.3 Deletion and Cycling

Many people, with prolonged viewing, switch back and forth between seeing a rabbit and a duck. This phenomenon, which we refer to as cycling, suggests that people may actively discard previously-held perceptions. To capture such behavior, we now allow the agent not only to add features with high fit, but also *delete* features with low fit.

We augment the model of Section 3 as follows. As before, in each period, the agent evaluates a feature $f$'s fit with the contents of working memory $W_t$. Now, however, the agent considers $f$ either for addition to or deletion from the knowledge set. Specifically, the agent may evaluate a feature $f \notin K_t$, in which case they add $f$ to the knowledge set if fit exceeds a threshold $\alpha$ ($\Phi(f, W_t) \geq \alpha$); or the agent may evaluate a feature $f \in K_t$, but with $f \notin W_t$, in which case they delete $f$ from the knowledge set if fit is below a threshold $\beta$ ($\Phi(f, W_t) \leq \beta$). We assume that the thresholds $\alpha$ and $\beta$ satisfy $0 \leq \beta < \alpha \leq 1$. As discussed in Section 3, if no picture is consistent with the "facts" ($Z(W_t, R) = 0$), the fit function is undefined; in this case, we assume that $f$ is neither added nor deleted. We also assume that the agent's initial knowledge $K_0$ is "axiomatic" and cannot be deleted.

The following proposition shows that cycling is now a possibility.

**Proposition 7.** *Under Assumption 1, cycling is possible. That is, for some* $(\alpha, \beta, L, P, C, K_0)$, *there exist features* $f$ *and* $f'$ *and a working memory sequence* $((W_0, f_0), (W_1, f_1), ...)$ *such that:*

- $f \in K_{t_1}$ *and* $f' \notin K_{t_1}$,
- $f' \in K_{t_2}$ *and* $f \notin K_{t_2}$,
- $f \in K_{t_3}$ *and* $f' \notin K_{t_3}$,

*for some* $t_1, t_2, t_3$ *with* $t_1 < t_2 < t_3$.

To gain intuition for Proposition 7, let us return to the rabbit-duck example and, for the purposes of this discussion, let us stipulate that the left-hand side of the picture is more duck-like while the right-hand side is more rabbit-like. Imagine the agent initially thinks the picture depicts a rabbit (i.e. "rabbit" is in the knowledge set), as shown in Figure 13(a). The agent might reconsider this view, deleting "rabbit" from knowledge, if they evaluate "rabbit" against features of the more duck-like, left-hand side.

Recall that when "rabbit" is in the knowledge set, it tends to block the adoption of "duck" (see Figure 12). But having eliminated "rabbit" from knowledge, the agent is in a position to add "duck"—perhaps by evaluating "duck" against features of the duck-like, left-hand side (Figure 13(b)).

With "duck" in knowledge in place of "rabbit," the stage is set for the process to work in reverse. The agent might remove "duck" from knowledge if they evaluate it against features of the more rabbit-like, right-hand side (Figure 13(c)); and with "duck" removed, the agent is in a position to add "rabbit" back again (Figure 13(d)).

Proposition 7 holds even in the absence of Assumption 1. In fact, this assumption makes the result more challenging to establish, as it restricts the set of feasible working memory sequences.

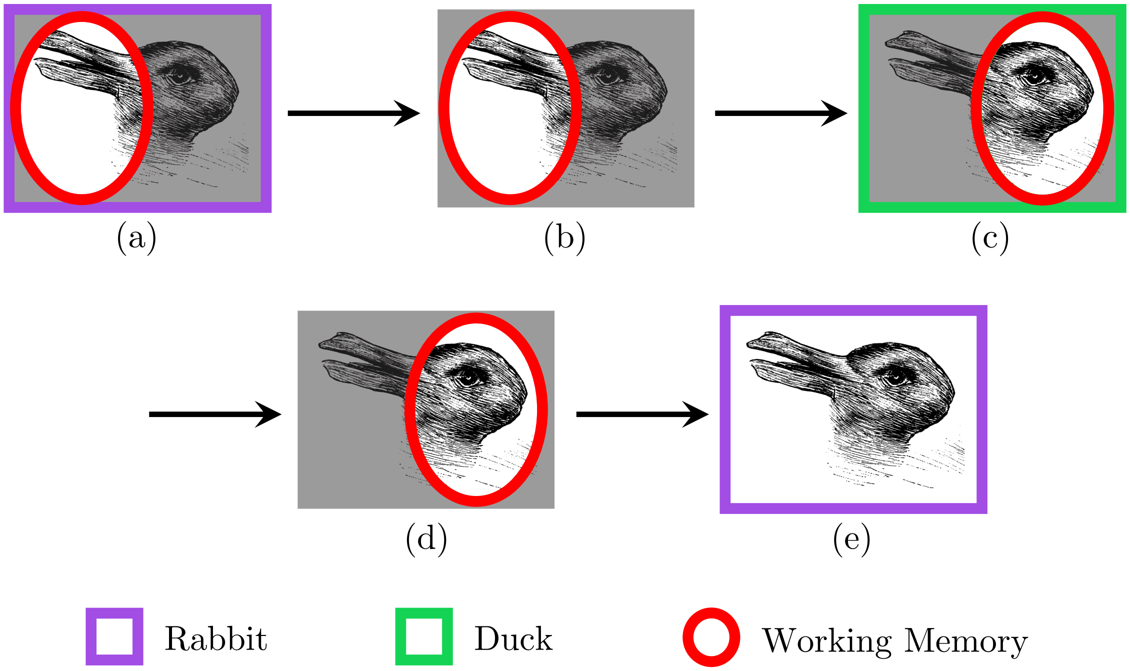


Figure 13: Example of Cycling

## 4.4 Mental Scaffolding

In Figure 13, the features “rabbit” and “duck” are both vulnerable to deletion. A natural question is whether there are mechanisms that might protect such features against deletion. Here, we show that the presence of a complementary feature in knowledge can be protective. For instance, “rabbit” and “rabbit ears” might protect each other, preventing the deletion of either from knowledge and blocking the adoption of an alternative interpretation of the picture (“duck” and “duck bill”). The following proposition captures this type of interplay.

**Proposition 8.** *Under Assumption 1, for some $(\alpha, \beta, L, P, C, K_0)$, there exist features $f$ and $f'$ that protect each other from deletion. Specifically, given any working memory sequence $((W_0, f_0), \ldots, (W_{t-1}, f_{t-1}))$:*

- *If $f, f' \in K_t$, then for any subsequent working memory sequence, $f, f' \in K_\tau$ for all $\tau > t$.*
- *If $f \notin K_t$ or $f' \notin K_t$, then there exists a subsequent working memory sequence such that $f, f' \notin K_\tau$ for some $\tau > t$.*

The reason two features—such as “rabbit” and “rabbit ears”—may protect one another is as follows. Suppose “rabbit” and “rabbit ears” are both powerful, simple features. Then, per Assumption 1, whenever the agent considers removing “rabbit ears” from knowledge, “rabbit” comes to mind—which protects “rabbit ears.” Effectively, the agent says: “these must be rabbit ears since this is a rabbit.” By the same token, whenever the agent considers removing “rabbit”, “rabbit ears” comes to mind—which protects “rabbit.”

We can think of “rabbit” (a feature of the full picture) as a high-level narrative that is supported by complementary sub-narratives—such as “rabbit ears” (a feature of the left-hand side)—that fit well with “rabbit”. We think of such sub-narratives as “mental scaffolding” that help stabilize and protect high-level narratives.

A stable perception—as opposed to an outcome where there is cycling—is less likely when the scaffolding is weaker. For instance, when "rabbit ears" is less complementary to "rabbit" (i.e. has less good fit)—or is a more complex feature—it does less to stabilize "rabbit." Overall, sets of simple, complementary features form stable scaffolds.[33]

## 4.5 Patchwork Quilts Revisited

Notice that deleting features can help the agent iron out inconsistencies in their interpretation of the picture. For instance, if the agent has a mostly "rabbit" interpretation of the picture's parts, they might delete an inconsistent duck-like feature with poor fit. Nonetheless, this ironing process does not eliminate patchwork quilts entirely. The following proposition formalizes this point.

**Proposition 9.** *Under Assumption 1, patchwork quilts can arise and be stable. That is, for some* $(\alpha, \beta, L, P, C, K_0)$*, there exists an initial working memory sequence* $((W_0, f_0), \ldots, (W_t, f_t))$ *such that:*

1. *There is a stable set of features* $F$ *in the knowledge set after time* $t$*:* $F \subseteq K_\tau$ *for all* $\tau > t$ *for any working memory sequence* $((W_{t+1}, f_{t+1}), \ldots)$*.*

2. *No picture is consistent with these features:* $\{P : f \text{ applies to } P \text{ for all } f \in F\} = \varnothing$*.*

The idea behind the proposition is as follows. To eliminate an inconsistency from knowledge, the agent must detect an inconsistency; and detecting an inconsistency may be difficult with limited working memory. To illustrate, consider again the patchwork quilt example where the agent thinks the picture: has more than two black pixels, an even number of black pixels, and a prime number of black pixels. If the agent evaluates

[33] A related point, that we do not explore further in this paper, is that stability is enhanced when the scaffold has good "coverage" (i.e. there are sub-narratives covering most regions of the picture). For instance, "rabbit ears" (a sub-narrative for the left-hand side) and "rabbit head" (a sub-narrative for the right-hand side) provide good coverage for "rabbit." In the absence of effective coverage, an antagonistic sub-narrative might be able to gain a toehold in an uncovered region—which can ultimately lead to the deletion of the main narrative. For instance, "duck head" might take root in the absence of "rabbit head," leading to the displacement of "rabbit."

one of these features—for instance "more than two black pixels"—against the other two, that feature will be deemed to have zero fit and it will be deleted. However, evaluating one feature against the other two requires storing all three in memory at once—which may not be possible with limited memory capacity.

This example demonstrates that detecting inconsistencies may require complex triangulation that surpasses the agent's cognitive capabilities. The Penrose Triangle (Figure 14)—a type of illusion known as an "impossible object"—provides a visual illustration of this concept. The three-dimensional object that the graphic depicts is geometrically impossible, yet people struggle to detect the impossibility because doing so requires simultaneously evaluating all facets of the object.

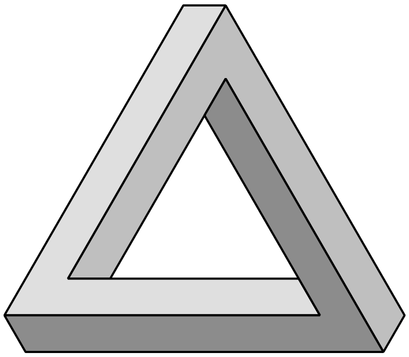


Figure 14: Penrose Triangle

## 5 A Choice Problem

So far, our focus has been on settings where the agent only faces a perception problem. They draw conclusions; but they do not *do anything* with those conclusions. Here, we extend the model to a choice-problem setting where the agent uses their conclusions to make choices.

Specifically, we consider a setting where the agent has all of the information needed to make an optimal choice. In other words, they have all of the pixels. However, to make a choice, the agent needs to *make sense* of that information. That is, they need to work out the *big picture*.

## 5.1 Setup

An agent has a utility function $u(a, o)$ over apples and oranges, where $a$ and $o$ denote quantities of apples and oranges respectively. The agent faces a choice between option 1, consisting of $a_1$ apples and $o_1$ oranges, and option 2, consisting of $a_2$ apples and $o_2$ oranges.

Initially, the agent knows the details of each option ($a_1$, $o_1$, $a_2$, and $o_2$); however, knowing these details is not the same as knowing which option they prefer. We have in mind that the agent only makes a choice when they put this information together and reach a conclusion about their preference. That is, they choose option $i$ when they conclude that $u(a_i, o_i) \geq u(a_j, o_j)$.

We apply our perception framework to model the agent's reasoning process about their preferences. The agent's choice problem can be represented as a picture $P$:

$$P = \begin{array}{r c} & \begin{array}{cc} \text{option 1} & \text{option 2} \end{array} \\ \begin{array}{r} \text{apples} \\ \text{oranges} \end{array} & \left[ \begin{array}{cc} a_1 & a_2 \\ o_1 & o_2 \end{array} \right] \end{array}$$

We permit $a_i$ and $o_i$ to take integer values between 0 and $z$, rather than just two values, to make the choice problem richer.

The agent reasons about the picture using the extrapolative procedure described in Section 3.1, where new features are evaluated against existing features in working memory and added if fit exceeds threshold $\alpha$.[34] The agent's initial knowledge set $K_0$ consists of the pixels of $P$ ($a_1$, $o_1$, $a_2$, and $o_2$). They choose option $i$ if they add a feature $f_i$ to the knowledge set corresponding to the event where $u(a_i, o_i) \geq u(a_j, o_j)$.

*Remark.* Notice that if we allow for deletion of features, the agent might add $f_i$ to knowledge, leading them to choose option $i$, and then later delete $f_i$ from knowledge, possibly replacing it with $f_j$. This corresponds to a case where the agent doubts or regrets a deci-

[34] The only difference from Section 3.1 is that pixels take more than two values; thus the number of possible pictures is $(z + 1)^4$ instead of $2^4$.

sion they have already made.

## 5.2 Analysis

Let us analyze the agent's decision problem under two simplifying assumptions. First, suppose the utility function has an additive form: $u(a, o) = a + o$. Second, suppose the agent's code only assigns short codewords to pixels, $f_1$, and $f_2$; all other features are too long to fit into working memory. This rules out the possibility of using chunking to reach decisions.

**Case 1: high working memory capacity.**

First, consider the case where the agent's working memory capacity ($L$) is sufficient to load all of the pixels into working memory ($W_t$) when considering $f_1$ or $f_2$ for addition to the knowledge set. If the agent loads all of the pixels and considers $f_i$, they add $f_i$ (and hence choose option i) if and only if $u(a_i, o_i) \geq u(a_j, o_j)$. Intuitively, because the agent has a large memory capacity, they are able to work out which option maximizes utility. Thus, provided the agent fully exploits their memory capacity, the classical assumption holds that the agent makes the utility-maximizing choice.

**Case 2: low working memory capacity.**

How does the agent reason when they cannot load all of the pixels into working memory? Consider two possible approaches the agent might take when they can only load two pixels (details of the analysis are in Appendix A.2).

*Focusing on one attribute.* The agent might fixate on a particular attribute of the choice problem—such as apples—just as agents in the work of BGS et al. selectively attend to attributes such as price or quality.

Suppose the agent attends to apples (i.e. puts $a_1$ and $a_2$ into working memory) when considering $f_i$. It is easy to show that the agent adds $f_i$ to knowledge—that is, chooses option $i$—if and only if $a_i - a_j$ exceeds a threshold $\theta$. Intuitively, if option $i$ yields sufficiently more apples than option $j$, the agent is willing to extrapolate to the conclusion "option $i$ yields more utility than option $j$."[35]

*Focusing on one option.* Alternatively, the agent might fixate on one option $i$ (i.e. by putting $a_i$ and $o_i$ into working memory). In that case, the agent satisfices in the sense of Simon: they choose option $i$ if it exceeds a "good enough" threshold. That is, if the agent considers $f_i$, they add it to knowledge if $u(a_i, o_i)$ exceeds a threshold $\eta$. Intuitively, if option $i$ yields sufficient utility, the agent is willing to extrapolate to the conclusion "option $i$ yields more utility than option $j$."[36]

## 5.3 Struggling to Choose

In order to make a decision, the agent needs a narrative about which choice is best. Absent such a narrative, they struggle to choose.[37] Constructing such a narrative is difficult when working memory is limited. For instance, suppose the agent fixates on apples. If $|a_1 - a_2| < \theta$, the agent will not be able to extrapolate to either $f_1$ or $f_2$, leaving them unable to pick an option. Intuitively, given that the agent can only attend to limited information, they are unable to muster sufficient confidence in either choice.

[35] Here, the agent chooses an option based on a particular attribute (e.g. apples). In a slightly richer version of the model, we can also capture situations where an agent rejects an option based on a particular attribute, and uses a sequential rejection process to ultimately make a choice. Tversky (1972) argues that this elimination process is a common practice in complex choice settings.

[36] The agent might also attend to the attributes of option $j$ when considering $f_i$. In this case, the agent adds $f_i$ to knowledge (i.e. chooses option $i$) if and only if the utility option $j$ yields is *below* a threshold: $u(a_j, o_j) \leq \eta'$.

[37] The challenge of decision-making in the absence of a guiding narrative is well-supported across disciplines. The work of David Tuckett and coauthors on conviction narratives emphasizes the role of storytelling in shaping confidence, suggesting that people construct narratives to navigate uncertainty and make complex choices with conviction (see Johnson et al. (2023)). Without such narratives, they argue, decision-making becomes fraught with doubt and hesitation. Shafir et al. (1993)'s notion of "reason-based choice" similarly highlights how a lack of coherent rationale complicates choice, as individuals struggle when they cannot find a unifying reason or story. Donald Davidson's theory of action is foundational here, proposing that intentional action depends on having reasons that make sense within a larger interpretive framework (see Davidson (1963)). Cognitive load theory (Sweller (1988)) further suggests that without an overarching narrative to integrate disparate information, cognitive burden increases, heightening the risk of analysis paralysis (Baumeister et al. (1998)). Additionally, Ricoeur (1992) and McAdams (1993) underscore the role of narrative in constructing a cohesive self, while Chang (2017)'s work on "hard choices" illustrates the difficulty of making decisions in the absence of clear evaluative criteria.

Besides working memory capacity, several other factors affect whether it is difficult to choose. A second factor is choice complexity, consistent with the empirical literature on choice overload. Increasing the number of options (e.g., three instead of two) or the number of attributes (e.g., apples, oranges, and bananas) raises the total number of pixels, making it harder for the agent to feel confident about any particular choice (see Appendix A.2 for details).[38] This finding echoes Iyengar and Lepper (2000)'s classic "jam study," which shows that consumers purchase more jam when presented with fewer options (six) than when offered a larger assortment (twenty-four).

Another factor is how similar attributes are across options. For example, consider an agent who focuses on a particular attribute (e.g., comparing $a_i$ to $a_j$). As $|a_i - a_j|$ shrinks, the options converge in attractiveness, making it harder for the agent to choose. This result aligns with a body of work showing that, when options are close in attractiveness, people engage in costly deferral, even though doing so is irrational (e.g. Tversky and Shafir (1992) and Dhar (1997)).[39,40]

What happens when agents struggle to choose? One possibility is that agents may go with a default—in effect, *not* making a choice. This might explain why defaults are so influential in complex contexts, such as retirement-plan selection (see Madrian and Shea (2001) and Thaler and Benartzi (2004)).

A second possibility is that agents might gradually lower the fit threshold $\alpha$ (i.e., they become more willing to extrapolate) until they reach a point where making a choice is feasible.[41] This is in line with a large experimental literature suggesting that difficult

[38] For instance, adding a third option raises the satisficing threshold $\eta$, making it less likely that the agent will consider any option "good enough." Intuitively, the more options there are, the less willing the agent becomes to deem the current option "the best."

[39] In Tversky and Shafir (1992)'s experiment, a rational agent might be more inclined to defer when the overall attractiveness of options declines, but not when options become closer in relative attractiveness. Nonetheless, they find that relative attractiveness matters.

[40] Another factor that affects whether it is hard to choose is the agent's code. For instance, suppose the agent has short codewords for the features "$a_i \geq a_j$" and "$o_i \geq o_j$." The ability to chunk such information enables the agent to deduce that $u_i \geq u_j$ even when $|a_i - a_j|$ and $|o_i - o_j|$ are both small—but only if there is a dominant option (i.e. $a_i \geq a_j$ and $o_i \geq o_j$). Note that, as an agent gains more experience with a certain type of choice problem, their coding scheme may become better adapted, rendering it easier to make decisions.

[41] The agent is able to make a choice if the fit threshold is sufficiently low. If the fit threshold is zero, all

choices take longer to make (e.g. Payne et al. (1993)).

## 6 Persuasion

Political actors frequently try to frame the narratives around their campaigns and around controversies (e.g. Entman (1993); Lakoff (2014)). An early television-era example is Richard Nixon's "Checkers" speech, delivered two months after his selection as Eisenhower's running mate in response to allegations of an improper $18,000 campaign fund. Opening by conceding that his honesty and integrity had been questioned and insisting that the best response to a smear "is to tell the truth," Nixon then detailed how every penny went to political expenses—not personal gain—and walked viewers through his modest household budget, from a rented apartment in Alexandria to his wife Pat's plain winter coat. The defining moment came when he revealed a gift from a supporter—a cocker spaniel named Checkers—and declared it the one gift he intended to keep. That simple, heartfelt anecdote shifted the focus, humanized Nixon, and swiftly defused calls to drop him from the ticket.

Standard economic models of persuasion, where a principal influences an agent solely by choosing what information to disclose (e.g. Kamenica and Gentzkow (2011)), overlook a crucial aspect of Nixon's rhetorical strategy: he did not change what voters learned so much as change how they understood it. In our model, an agent's beliefs depend both upon the information they receive and on how they *interpret* it, opening up new channels of influence.

### The Role of Timing

One new channel of persuasion is manipulating the time at which information is released. Consistent with a psychological literature on primacy effects (e.g. Asch (1946)), whether an agent sees good information earlier or later affects the narrative they ultimately adopt.

choices are acceptable; thus, the agent effectively chooses at random.

To illustrate, consider the following example (further details of which are given in Appendix A.3). There is a representative voter choosing between candidate A and candidate B. Candidate B's type is known, so the voter's choice will depend on their assessment of candidate A, whose type consists of a vector of $n$ pixels:

$$\begin{bmatrix} p_1 & p_2 & p_3 & \cdots & p_n \end{bmatrix},$$

with $p_i \in \{\text{good}, \text{bad}\}$ and $n > 4$. We will refer to candidate A as a "good egg" if there are at most two bad pixels and a "bad egg" if there are at most two good pixels.

In the voter's code, the only features with codewords short enough to fit in working memory are pixels, "good egg," and "bad egg"; any combination of these features can be loaded into working memory. The voter extrapolates with addition and deletion thresholds $\alpha$ and $\beta$. They vote for candidate A if they think candidate A is a "good egg," and they vote for candidate B if they think candidate A is a "bad egg."

Suppose that only four pixels are revealed to the voter: two of which are good and two of which are bad. Candidate A cannot block these pixels from being revealed but can control the sequence of revelation. Candidate A decides which pixels are revealed (i.e. added to knowledge) in round 1 and which are revealed in round 2. In each round, voters make extrapolations over many steps until they settle on a stable knowledge state (if one exists).

For some range of extrapolation thresholds, we find the following (see Appendix A.3 for further details).

*Case 1: the good pixels are revealed in round 1 and the bad pixels are revealed in round 2.* The voter can extrapolate to "good egg" off of the two good pixels but not to "bad egg." After the voter has extrapolated to "good egg," the revelation of the two bad pixels in the next round is not sufficient to delete the "good egg" narrative. Thus, candidate A wins when the good pixels are revealed first.

*Case 2: the bad pixels are revealed in round 1 and the good pixels are revealed in round 2.* The voter can extrapolate to “bad egg” off of the two bad pixels but not to “good egg.” After the voter has extrapolated to “bad egg,” the revelation of the two good pixels in the next round is not sufficient to delete the “bad egg” narrative. Thus, candidate B wins when the bad pixels are revealed first.

Consequently, candidate A prefers that their positive traits be revealed first. In contrast, candidate B would rather have the voter initially see candidate A’s negative traits. Similar to the rabbit–duck illusion, two narratives—“good egg” and “bad egg”—are possible; the information that is disclosed early determines which narrative voters adopt.

### Suggesting Narratives

Another way in which a persuader might influence an agent is by *suggesting a narrative*. We have in mind that suggesting a narrative involves affecting what the agent attends to (i.e. loads into working memory). Specifically, suppose the persuader suggests a narrative $f$ to the agent at time $t$ and points out a set of facts $F$ that the agent already accepts (i.e. $F \subseteq K_t$). This suggestion prompts the agent to consider adopting $f$ at time $t$, while storing facts $F$ in working memory (i.e. $W_t = F$).

Politicians like to be the first to suggest a narrative—because the initial narrative has an outsize impact on what voters believe. Indeed, they will sometimes release damaging information themselves, not to harm their own standing but to deny opponents the chance to frame the story before them (see Arpan and Roskos-Ewoldsen (2005)).

To illustrate, consider the following elaboration of the previous example (with further details again in Appendix A.3). Suppose there is a negative story about candidate A (e.g. concerning improper use of campaign funds or an illicit affair). Candidate A is aware of the story and the opposing candidate (B) is aware of the story with probability $p$. The story consists of a vector of $n > 4$ pixels where, once again, each pixel is either good or bad. The good pixels correspond to mitigating factors (e.g., if the story involves an illicit affair, mitigating factors might include its brevity and the politician’s reconciliation with their spouse). The story is “forgivable” if there are at most two bad pixels and “unforgivable” if there are at most two good pixels. If the story comes out, the voter only sees four of the $n$ pixels: two good pixels and two bad.

The probability that candidate A wins the election is:

$$Prob(\text{A wins}) = \begin{cases} 1, \text{ if the story does not come out} \\ q, \text{ if it comes out and the voter deems it “forgivable”} \\ 0, \text{ if it comes out and the voter deems it “unforgivable.”} \end{cases}$$

The candidates alternate turns, with candidate A moving first. On each turn, the active candidate may release the story if they are aware of it. Once the story has been released, the active candidate can also suggest a narrative about it—“forgivable” or “unforgivable”—and direct attention to specific aspects of the story (for example, the good pixels or the bad pixels). Candidates may also take the voter through two working-memory steps regarding the story, first having them consider rejecting the existing narrative, and then having them consider adopting a new one (for instance, rejecting “forgivable” and adopting “unforgivable”). After this initial suggestion, the voter makes further extrapolations over many steps, and the candidate’s turn then ends.

It is easy to show that, for the same extrapolation thresholds as in the previous case, the first candidate to release the story determines the narrative the voter ultimately adopts. Candidate A can release the story first, suggest the “forgivable” narrative to the voter while directing their attention to the good pixels (the mitigating factors); the voter then adopts this narrative and candidate B cannot subsequently get the voter to switch to the “unforgivable” narrative. Likewise, if candidate B releases the story first, they can suggest the “unforgivable” narrative to the voter while directing their attention to the bad pixels (the negative aspects of the story); the voter then adopts this narrative and candidate A cannot subsequently get the voter to switch to the “forgivable” narrative.[42]

[42] Here, we allow the candidate to suggest a narrative ($f$) *and* focus the voter’s attention on certain facts ($W_t$). More generally, the voter could be persuaded through one of these channels alone: that is, simply

Thus, candidate A faces a choice whether to "get ahead of the story." If they release the story first, they can get the voter to adopt their preferred narrative ("forgivable") and they win with probability $q$. Alternatively, they can remain silent, in which case candidate B releases the story if they are aware of it (i.e. with probability $p$) and steers the voter to their preferred narrative ("unforgivable"). It follows that candidate A releases the story if $p + q > 1$.

**Simple Narratives**

A further implication of the model is that persuaders tend to be more effective when they suggest simple, powerful narratives since, as discussed earlier (Section 3.2.3), these narratives are easier to keep in mind.

All politicians know the importance of simple, powerful messages. Ronald Reagan's famous line "Government is not the solution to our problem, government is the problem" and Franklin Roosevelt's pronouncement "The only thing to fear is fear itself" resonate even today.

One corollary is that simple, powerful narratives can be quite persuasive even when they are *false*. Sometimes the truth is complex (i.e. agents may lack short codewords); in such cases, false narratives that are simple and powerful may be hard to counteract. As Jonathan Swift wrote: "Falsehood flies, and the Truth comes limping after it."[43]

# 7 Discussion

Here, we discuss a variety of applications and possible extensions of the model.

## 7.1 Multiple Observations and Model-Building

One interpretation of our framework could be that $P$ is not a single observation but rather a collection of (potentially related) observations. An agent who is presented with such a

by suggesting a narrative, or simply by focusing attention on certain facts.

[43] A modern version of this adage is Brandolini's (2013) law: "the amount of energy needed to refute bullshit is an order of magnitude bigger than to produce it."

collection might be particularly interested in the relationships between observations; that is, they may seek to perceive common features that are shared across all observations in the collection. For instance, the agent might perceive that each observation in the collection depicts a face. The agent's knowledge of common features is, in effect, the agent's model of that collection.

Consider a concrete example. In the picture shown below, each column describes an experiment undertaken by the agent, where the agent took a set of actions ($a_{1i}, ..., a_{ni}$) and observed a set of outcomes ($o_{1i}, ..., o_{mi}$):

$$P = \begin{bmatrix} a_{11} & \dots & a_{1k} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nk} \\ o_{11} & \dots & o_{1k} \\ \vdots & \ddots & \vdots \\ o_{m1} & \dots & o_{mk} \end{bmatrix}.$$

Some of the features of $P$ describe patterns that are common to each column. For instance, one feature the agent might perceive would be that $a_{1i} = o_{1i}$ for all $i$. This feature is effectively a model of the world in which a particular action ($a_{1i}$) determines a particular outcome ($o_{1i}$). Another feature the agent might perceive would be that $o_{1i} = o_{2i}$ for all $i$. This is effectively a model of the correlation between certain types of outcomes.

Notice that the agent might initially observe all of the pixels of $P$, in which case a feature like $a_{1i} = o_{1i}$ is a purely *descriptive model*. However, some actions/outcomes might be unobserved, in which case $a_{1i} = o_{1i}$ might serve as a *predictive model*.

## 7.2 Back-and-Forth Communication

An important idea in organizational economics is that boundedly rational individuals can achieve superior outcomes when working together (see Simon (1947)). A version of our model with multiple agents speaks to this idea; moreover, it makes sense of why it is important for agents to have back-and-forth communication.

Consider, for instance, a team-theoretic setting where two agents have different codes. They are presented with the same picture $P$ and have the same initial knowledge, consisting of all of the pixels of $P$. Both agents are purely deductive. Suppose both agents are able to deduce something but not everything about the picture on their own; and because of their different codes, the agents deduce different things. Let $F_1^0$ and $F_2^0$ denote the features that are deducible for agents 1 and 2 respectively.

Clearly, the agents benefit from sharing their deductions: agent 1 can add $F_2^0$ to their knowledge set and vice-versa. However, the benefits do not stop there. Notice that agent i, having added features from $F_j^0$ to their knowledge set, may be able to make further deductions. Let $F_i^1$ denote the additional deducible features for agent i. The agents can communicate these additional deductions; and perhaps even more deductions will result ($F_i^2, F_i^3, ...$).

We see then that communicating back-and-forth can generate continual revelations. This captures a feature of communication that is intuitive, yet absent from existing models. In models where agents simply have different initial information, they share what they know and there is no further reason to communicate.

Notice that when the agents' codes are exactly the same, communication does not yield fresh insights ($F_1^0 \cup F_2^0 = F_1^n \cup F_2^n$ for $n > 0$)—since each agent already sees what the other sees. Likewise, if the agents' codes are extremely different, communication does not yield fresh insights. Intuitively, if the agents communicate, agent 1 shares features that are simple for them but complex for agent 2. Thus, agent 2 cannot put agent 1's insights to use to learn more about the picture. However, if the agents codes are different—but not too different—communication does yield fresh insights. Thus, communication is most fruitful in this intermediate case.

## 7.3 Categorization

The psychological literature on categorization, starting with the work of Eleanor Rosch and colleagues, highlights a fascinating phenomenon (see, for example, Rosch (1973)). People normally perceive categories in binary terms: an item either belongs or not. At the same time, some items are considered more exemplary than others. For instance, a robin is generally perceived as a more exemplary "bird" than an ostrich.

In fact, the most exemplary items—rather than being the most common or average—are often exaggerated forms. Young boys often dress up as soldiers and young girls as princesses precisely because these exaggerated images are particularly exemplary of the categories "male" and "female."

Our model speaks to these findings. Features are binary in the model: a feature $f$ either applies to a picture $P$ or not. From a binary feature $f$, however, we can classify pictures in non-binary terms as more or less exemplary of feature $f$. Specifically, let us say that an agent considers a particular picture $P$ more exemplary of feature $f$ if it is easier for the agent to extrapolate to $f$ (easier in the sense of lower working memory requirement $L$ or stricter fit threshold $\alpha$). For instance, an agent might find it easier to extrapolate to "bird" from a picture of a robin than a picture of an ostrich.

Notice that this allows for an alternative—and somewhat more appealing—definition of smiley faces. Previously, we defined "smiley-face" in binary terms: pictures were smiley-faces or they were not (see Figure 11). We might instead use a particular feature $f^*$ (e.g. consisting of one or several smiley-face pictures) to classify pictures as more or less exemplary smiley-faces.[44]

We might ask what pictures are most exemplary of a category. For instance, which pictures most *scream* "bird"? One might guess that the picture that most screams "bird" *is* a picture of a bird (i.e. feature $f^*$ applies to the picture most exemplary of feature

[44] Rosch argues that people assess how well an item fits a category based on its similarity to a prototype. Therefore, in the case of "smiley-face," feature $f^*$ would be a prototypical smiley-face. Medin and Schaffer (1978), by contrast, suggest that people compare against existing items in the category; these items serve as exemplars. Thus, in their view, $f^*$ consists of all items already classified as smiley-faces.

$f^*$). However, this is not necessarily the case. In fact, the birdiest picture may be highly exaggerated or abstracted: Picasso's take on a bird rather than a photograph. Note that the exact nature of this abstraction will depend critically upon the agent's code, as the code determines the nature of the extrapolation process.[45]

A classic experiment by the biologists Niko Tinbergen and Albert Perdeck illustrates the importance of exaggeration and abstraction (see Tinbergen and Perdeck (1951)). To induce a parent to regurgitate food, herring gull chicks peck at a red spot on the parent's bill (see Figure 15(a)). Tinbergen and Perdeck ran a series of experiments to test the pecking behavior of these chicks. They found that the chicks peck just as much at a model of an adult gull's head, or a model of the beak alone. However, most surprisingly, they found that a red rod with three white stripes (see Figure 15(b)) induced 25 percent more pecks from the chicks than any of the other alternatives. In other words, the abstracted beak—which wasn't a beak at all—was most exemplary to the chicks.

(a) Herring gull.

(b) Red Rod.

Figure 15: Tinbergen and Perdeck's Experiment

## 7.4 Latent Structure

There are many systems where people observe some aspects of the system, but they fail to see all of its workings. For instance, before the invention of the microscope, people could observe infectious illnesses but they were unable to see the germs causing those diseases.

[45] Another approach to exaggeration is offered by Bordalo et al. (2016a). They develop a model of stereotypes based on Kahneman and Tversky's representativeness heuristic, where agents overweight the frequency of relatively common types.

In such settings, people not only form views of what they are seeing; they also form views regarding these latent structures. Take Darwin's theory of natural selection, for instance. He compiled a vast array of observations—from the beaks of finches in the Galapagos to the fossils of giant armadillos in South America—but he could not directly observe the evolution of species over millions of years. He managed nonetheless to take these observations and form a view of the latent structure. Indeed, an immense number of scientific theories—from string theory to the structure of DNA—require inferences about such structure.

It turns out that it is relatively easy to extend the model to allow agents to perceive latent structure. We illustrate using an example from the realm of visual perception.

**Perceiving a third dimension.**

People often look at two-dimensional images and interpret them as three-dimensional scenes. A classic example is the Necker cube (Figure 16). We can think of the three-dimensional cube as the inferred latent structure.

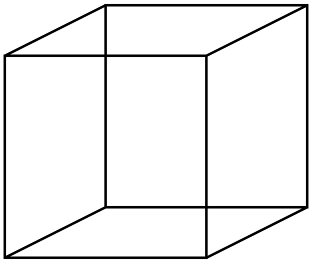


Figure 16: The Necker Cube

Let us consider a simple tweak to the model that captures the inference of latent structure. Suppose the agent thinks that the two-dimensional picture $P$ they are presented with is the projection of a three-dimensional array $\tilde{P}$ onto a two-dimensional plane (akin to a photograph). Formally, let $P = G(\tilde{P})$, where $G : \mathbb{R}^3 \to \mathbb{R}^2$ is a projection of a three-dimensional space onto a two-dimensional plane.

Observe that for any feature $f$ of $P$, there is a corresponding feature $\tilde{f}$ of $\tilde{P}$, where

$\tilde{f} = \{\tilde{P}' : f \text{ applies to } G(\tilde{P}')\}$. Let $K_0$ denote the agent's initial knowledge of picture $P$, consisting of pixels. Let $\tilde{K_0}$ denote the corresponding features of $\tilde{P}$. The tweak we make to the model is that we assume that the agent deduces/extrapolates on the three-dimensional space, starting from $\tilde{K_0}$—rather than on the two-dimensional space.

A consequence of projecting three dimensions down to two is that, if the agent is purely deductive, they will not be able to tell what image they are looking at in three-dimensional space—even if they can deduce everything in two-dimensions. Ambiguity regarding the latent structure remains. For instance, a purely-deductive agent could not rule out that the image in Figure 16 is flat.

However, when the agent extrapolates, they may reach a firm view despite this underlying ambiguity. The Gestalt psychologists argued that people employ a variety of visual strategies when looking at images, which help them—among other things—convert two-dimensional images into three dimensions. These strategies include, for instance, looking for symmetries, converging lines (indicative of depth), and objects obscuring other objects (also indicative of depth). Recall that agents in our model are particularly likely to extrapolate towards interpretations coded as simple. We might think of these strategies identified by the Gestalt psychologists as features that agents tend to code as simple which, in turn, play a key role in determining how they resolve ambiguity regarding latent structure.[46]

## 7.5 Occam's Razor

Occam's razor is a principle that asserts that simple explanations are generally the best ones. One of the standard justifications for this principle is that the failure to favor simple explanations can lead to overfitting. Intuitively, a complex model—with many degrees of freedom—has more wiggle room to find an explanation that fits the data well but is

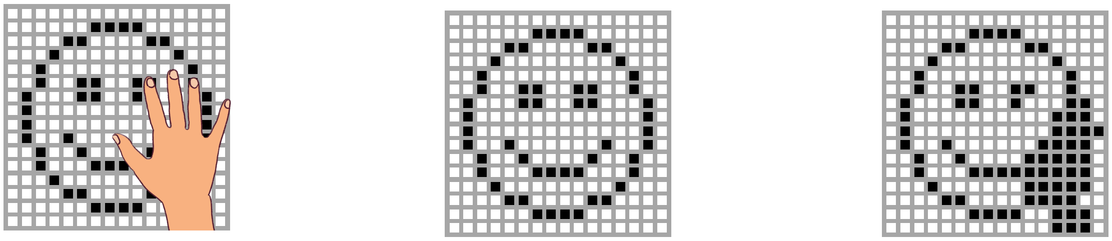
[46] The perception of motion can also be accounted for in terms of latent structure, which in this case relates to whether objects across pictures are considered to be the same or different. For instance, if a dot in picture 1 is viewed to be the same as a dot in picture 2, this would lead to the perception of a moving dot between pictures; by contrast, if the dots are viewed to be different, this would lead to the perception of one dot disappearing and another dot appearing.

ultimately incorrect.[47]

Notice that the fit function we chose in Section 3.1 does, in fact, take into account this overfitting concern. To illustrate, suppose we define a feature $f$'s "explanatory power" as the log share of pictures with feature $f$ among those that fit the facts $W_t$ (i.e. $\frac{\log Z(W_t \cup \{f\}, R)}{\log Z(W_t, R)}$). Fixing the explanatory power of $f$, it is less likely to be adopted (i.e. $\Phi(f, W_t)$ is lower) when there are fewer facts (i.e. $\frac{\log Z(\varnothing, R)}{\log Z(W_t, R)}$ is smaller). In other words, when there are only a few facts—which raises the chance feature $f$'s explanatory power is purely due to chance—the bar for adoption is greater.

There is another important way in which Occam's razor comes into play in our framework, however. Recall that agents are more likely to adopt explanations that they *code as simple*, since these explanations are easier for them to think about.

To illustrate, consider again the following picture partially obscured by a hand (Figure 17(a)). The extrapolation the agent makes regarding the unseen pixels will depend in large measure upon their code. In particular, they will be more likely to extrapolate to a smiley-face (Figure 17(b)) than some alternative with the same fit (Figure 17(c)) if they code smiley-faces as simple.



(a) Initial Observation

(b) Extrapolation 1

(c) Extrapolation 2

Figure 17: Two Extrapolations with the Same Fit

Agents' codes—and hence, their views of what is simple—may differ. This, in turn, may lead to differences in the extrapolations they make. In fact, one could experimentally

[47] For instance, versions of Occam's razor can be derived within a Bayesian model selection framework. Two classic formalizations that come from such an approach are the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC), which reward models for fitting the data well but also penalize them for having more degrees of freedom.

test this hypothesis—for instance, by seeing whether novice and expert chess players, who appear to have different codes, extrapolate differently.[48]

## 8 Conclusion

The Gestalt psychologists' observation that seeing the parts is different from seeing the whole has become a foundational idea in cognitive psychology and neuroscience. Integrating information, and developing an understanding of what it means, is seen as one of the key challenges our brains face. Computer science and information theory are also based on the idea that processing information is not straightforward. Economics, by contrast, typically treats the information processing problem as a black box.

This paper builds a framework that helps bridge economics with these other disciplines. We assume that agents have unlimited ability to store information—unlimited hard-drive space, so to speak. However, they have limited working memory (akin to RAM). Working memory is the place where agents integrate information and develop a big-picture understanding of what it means. We show that limited working memory bounds agents' ability to draw conclusions.

We also explore, in this context, the use of extrapolation by agents. On the one hand, extrapolation allows agents to reach further than they can with deduction alone, potentially reaching more correct conclusions. On the other hand, extrapolation introduces the possibility of wrong conclusions.

This paper leaves a number of unanswered questions. There are aspects of the reasoning process that require further exploration. How, for instance, are the contents of the

[48] If this hypothesis is confirmed, an additional empirical question one might ask is whether it is simply differences in cognitive load that affect which explanations agents adopt. For instance, the agent's cognitive load is likely to be considerably lower if they are asked to choose between two potential extrapolations such as Figures 17(b) and (c) rather than generate an extrapolation from scratch. We suspect, even if agents are given a simple choice, they will tend to extrapolate to Figure 17(b) over (c), suggesting that they may, ceteris paribus, have a preference for simple explanations. Notice that adding a penalty to the fit function for complexity would be a simple way of capturing such a preference (analogous to the penalty for more degrees of freedom imposed in the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC).

agent's working memory determined? The work of BGS et al. suggests that there are bottom-up processes at work (i.e. the salience of information) while the literature on rational inattention emphasizes the role of top-down processes (i.e. people make decisions regarding what deserves attention).

Our focus in this paper has primarily been on the role of limited working memory. However, the paper also brings to light the importance of limited cognitive ability (which we can think of as the agent's clock speed at performing deductions/extrapolations). When cognitive ability is limited, it matters not only *what* the agent is able to conclude but also *how quickly* they are able to reach conclusions.

# Appendix

## A.1 Proofs

*Proof of Proposition 1.* Part 1: suppose that at $t = 0$, the agent loads every $1 \times 1$ feature into working memory. (Recall that there are $M \times N$ such features, and each has a length-one codeword.) Then every feature of the picture is immediately deduced, so that $K_1$ consists of the set of all features of $P$, as claimed.

Part 2: observe that if $L = 1$, then $W_t$ in each period is either empty or consists of a single $1 \times 1$ feature. No new features can be deduced from a single $1 \times 1$ feature $f$, except for features equivalent to $f$. It follows that any feasible knowledge set $K_t$ consists only of features that are equivalent to features already in $K_0$. The result follows.

Part 3: consider the following example. $P$ is a $1 \times 5$ picture where every pixel is white ("all white"). Pattern $q$ consists of a single $1 \times 3$ "all white" submatrix and has a length-two codeword, while $q'$ consists of a single $1 \times 4$ "all white" submatrix and has a length-three codeword. No other pattern has a length-two codeword. Working memory capacity is $L = 3$.

Observe that any feature of the form $(q', R')$ cannot be deduced in one step; but it can be deduced in two steps by first learning a smaller feature of the form $(q, R)$ where $R$ is a subregion of $R'$. Furthermore, the full picture cannot be deduced. □

*Proof of Proposition 2.* Consider the following example. Suppose $L = 8$. Picture $P$ is a $4 \times 8$ "checkerboard": each pixel $p_{ij}$ is black if $i + j$ is even and white otherwise. The following patterns will be relevant (see Figure 18):

- $q_1$ consists of the two $1 \times 8$ "alternating" sub-matrices "black-white-black-…-white" and "white-black-white-…-black."

- $q_2$ is the set of $4 \times 8$ matrices where each $1 \times 8$ row of the matrix matches an element of $q_1$.

- $q_3$ is the $4 \times 1$ submatrix "black-white-black-white."

- $q_4$ is the $4 \times 7$ submatrix corresponding to the first seven columns of $P$.

Further, suppose that patterns $q_1$ and $q_3$ are assigned to length-two codewords, that $q_2$ is assigned to a length-six codeword, and that there are no other patterns with codeword length $\leq 8$.

For this proof, we will identify a region by the set of columns it covers; for instance, $R_{3-6}$ will refer to the region consisting of all pixel locations in the four middle columns.

The feature $(q_4, R_{1-7})$ applies to $P$; call it $f_4$. Similarly, we label the feature $(q_2, R_{1-8})$ as $f_2$. We claim that $f_4$ can only be deduced from a combination of $f_2$ and some feature based on pattern $q_3$. Observe that given $L = 8$, no combination of features that excludes

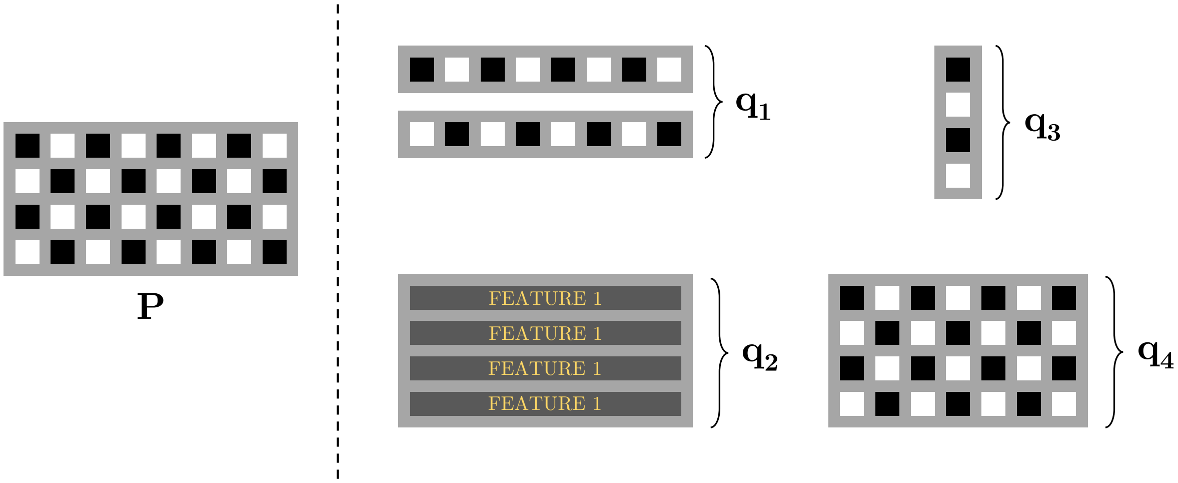


Figure 18: Proof of Bottom-Up/Top-Down Deduction

$f_2$ can deduce $f_4$. Further, if $f_2$ is loaded in working memory, there are only two units of working memory capacity remaining, and no two additional pixels nor any feature based on $q_1$ can suffice to pin down $f_4$. This establishes the claim.

On the other hand, there is a working memory sequence that eventually deduces $f_4$ from $f_2$ and some feature based on $q_3$. Specifically: in each of the first four periods, the agent loads all eight pixels from row $i$ into memory and deduces the feature $f'_i = (q_1, R'_i)$ where $R'_i$ is the region consisting of all pixel locations in the $i$-th row. In the fifth period, the agent loads each of $\{f'_1, \ldots, f'_4\}$ into memory and deduces $f_2$. In the sixth period, the agent loads all pixels from the first column into memory and deduces the feature $(q_3, R_1)$, which we label as $f_3$. In the seventh and final period, the agent loads $f_3$ and $f_2$ into memory and deduces $f_4$. It follows that $f_4$ can be, and can only be, deduced top-down.

Now consider the feature $(q_3, R_8)$, which we label as $f_5$. It can be deduced only from pixels (i.e. bottom-up). □

To establish Proposition 3, we will first state and prove two lemmas.

**Lemma 1.** *Consider a $1 \times 6$ grid. Let task $\tau = (F, P)$, where $F$ consists of all $64$ possible $1 \times 6$ pictures and $P$ is any given picture. (In words: the agent's task is to identify picture $P$ perfectly.) Then $L_{\min}(C^*(\tau), \tau) \leq 3$.*

*Proof of Lemma 1.* Let the pixels of $P$, from left to right, be $[p_1 \ldots p_6]$. For $i \in \{1, \ldots, 6\}$, let $q_{1\ldots i}$ be the $1 \times i$ pattern consisting of the matrix $[p_1 \ldots p_i]$. Consider a code where the patterns $q_{1\ldots 2}, q_{1\ldots 3}, q_{1\ldots 4}, q_{1\ldots 5}$ are assigned to length-two codewords. For $i \in \{1, \ldots, 6\}$, define $f_{1\ldots i} = (q_{1\ldots i}, R_{1\ldots i})$ where $R_{1\ldots i}$ is the region consisting of the first $i$ pixel locations, from left to right, of the matrix. Notice that learning $f_{1\ldots 6}$ is equivalent to identifying $P$. And, denote $f_i = (\{[p_i]\}, R_i)$ to be the feature where pixel $p_i$ applies to the region $R_i$

consisting of the $i$-th pixel location. Then picture $P$ can be identified with the working memory sequence.

$$W_t = \{f_{1\dots(t+1)}, f_{t+2}\} \text{ for } t \in \{0, \dots, 4\}.$$

Specifically, in each period $t \in \{0, \dots, 3\}$, the agent deduces $f_{1\dots(t+2)}$ from $f_{1\dots(t+1)}$ and $f_{t+2}$, and thus deduces $f_{1\dots6}$ at $t = 4$. $W_0$ takes up two units of working memory, while $W_1, \dots, W_4$ each take up three units of working memory. It follows that $L_{\min}(C^*(\tau), \tau) \leq 3$.

□

**Lemma 2.** *Consider a* $1 \times 6$ *grid. Let task* $\tau_S = (F, \mathcal{P})$ *where* $F$ *and* $\mathcal{P}$ *both consist of all* 64 *possible pictures. (In words: the agent's task is, given any* $1 \times 6$ *picture, to identify the picture exactly.) Then* $L_{\min}(C^*(\tau_S), \tau_S) \geq 4$.

*Proof of Lemma 2.* Assume towards a contradiction that there exists a code $C$ under which the task $\tau_S$ can be achieved with working memory capacity $L = 3$. Observe that with $L = 3$, any deductive step (where at least one new feature is deduced) must involve loading into working memory either (a) two or three $1 \times 1$ features, or (b) one $1 \times 1$ feature and one larger feature (with a length-two codeword).

Index the 64 possible pictures as $P_1, \dots, P_{64}$.

We say that pattern $q$ is *almost complete* if there exists a region $R$ (that matches $q$'s dimensions) and a $1 \times 1$ feature $f'$ such that $(q, R)$ and $f'$ together identify some picture $P$. In this case, we say that *$q$ almost identifies $P$*.

We claim that any almost-complete pattern can almost identify a maximum of twelve distinct pictures. To see why, note that any almost-complete pattern $q$ must be either (i) a $1 \times 5$ pattern consisting of a single $1 \times 5$ matrix $[\,p_1\, p_2\ \ p_3\, p_4\, p_5\,]$, or (ii) a $1 \times 6$ pattern. In case (i), one can see that exactly four distinct $1 \times 6$ pictures can be identified by the combination of $q$ and some $1 \times 1$ feature:

$$[\,b\, p_1\, p_2\ \ p_3\, p_4\, p_5\,] \quad [\,w\, p_1\, p_2\ \ p_3\, p_4\, p_5\,] \quad [\,p_1\, p_2\ \ p_3\, p_4\, p_5\, b\,] \quad [\,p_1\, p_2\ \ p_3\, p_4\, p_5\, w\,]$$

where $b$ represents black and $w$ represents white.

In case (ii), observe to start that there is a unique $1 \times 6$ feature $f$ associated with $q$. Further, there are only twelve $1 \times 1$ features, and thus no more than twelve ways to pair $f$ with a $1 \times 1$ feature to uniquely identify a picture. In either case, it follows that $q$ can almost identify at most twelve distinct pictures.

Consider any picture $P$, and consider a working memory sequence such that $P$ was deduced at the end of period $t$. Notice that in period $t$, to have deduced $P$, the agent must have loaded an almost-complete pattern $q$, together with a $1 \times 1$ feature, into working memory in period $t$. Given that $L = 3$, $C(q)$ must have length two (given that only $1 \times 1$-patterns have length one). However, there are only four length-two codewords, and thus at most four corresponding almost-complete patterns that the agent can use in identifying pictures. As each almost-complete pattern can almost identify at most twelve distinct pictures, the agent can deduce at most 48 distinct pictures. This is strictly less than the entire set of 64 pictures, and establishes our contradiction.

□

*Proof of Proposition 3.* Consider a $1 \times 6$ grid. Enumerate the 64 possible $1 \times 6$ pictures as $P_1, \ldots, P_{64}$, in any arbitrary order. Consider a series of tasks $\tau_i = (F, \mathcal{P}_i)$, $i = 1, \ldots, 64$, where

$$\mathcal{P}_i = \{P_1, \ldots, P_i\} \text{ and } F \text{ consists all all 64 possible } 1 \times 6 \text{ pictures.}$$

Let $L_i = L_{\min}(C^*(\tau_i), \tau_i)$ be the minimum working memory required to achieve task $\tau_i$.

Observe that $L_1 \leq 3$ (by Lemma 1); that $L_i$ is weakly increasing in $i$; and that $L_{64} \geq 4$ (by Lemma 2). Let $j = \min\{i : L_i \geq 4\}$, and let $\tau_j' = (F, P_j)$. Assume towards a contradiction that $C^*(\tau_{j-1}) \cap C^*(\tau_j')$ contains at least one element $C$.

We make two observations about $C$ here. First, by definition of $j$, each picture in the set $\mathcal{P}_{j-1}$ can be identified under $C$ with working memory constraint $L = 3$. Second, it follows from Lemma 1 that $P_j$ can also be identified under $C$ with working memory constraint $L = 3$. Combining these two observations, every picture $P_i$ with $i \in \{1, ..., j\}$ can be exactly identified using code $C$ and working memory constraint $L = 3$; it follows that

$$L_{\min}(C^*(\tau_j), \tau_j) \leq L_{\min}(C, \tau_j) \leq 3.$$

But this is a contradiction: by definition of $j$, we must have $L_{\min}(C^*(\tau_j), \tau_j) \geq 4$. We conclude that $C^*(\tau_{j-1}) \cap C^*(\tau_j') = \varnothing$, and thus that the Proposition holds. □

*Proof of Proposition 4a.* Fix $\alpha'$ and $\alpha'' \leq \alpha$. Consider any working memory sequence $((W_0, f_0), (W_1, f_1), \ldots)$ that is feasible given threshold $\alpha'$. We claim that this working memory sequence is also feasible under threshold $\alpha''$, and thus that every feature that is extrapolated under this working memory sequence given $\alpha'$ is also extrapolated under $\alpha''$.

For each $t$, if $f_t$ can be extrapolated given $W_t$ under $\alpha'$, it can also be extrapolated under $\alpha''$ (because the fit requirement is relaxed). It follows by induction on $t$, starting from $t = 0$, that sequence $((W_0, f_0), \ldots, (W_t, f_t))$ is feasible under $\alpha''$ if it is feasible under $\alpha'$. The claim, and part 1 of the Proposition, follows.

Observe further that when $\alpha = 1$, $f_t$ is extrapolated under $W_t$ if and only if $f_t$ can be deduced from $W_t$. The following claim then follows by induction on $t$: for any feasible working memory sequence $((W_0, f_0), (W_1, f_1), \ldots)$, each $W_t$ consists only of features that apply to $P$. Part 2 of the Proposition follows. □

*Proof of Proposition 4b.* The proof of Part 1 of the proposition is identical to that of Part 1 of Proposition 4a.

For Part 2, observe that when $\alpha = 1$, $f_t$ is extrapolated under $W_t$ if and only if $f_t$ can be deduced from $W_t$. The following claim then follows by induction on $t$, with $t = 0$ as the base case: for any feasible working memory sequence $((W_0, f_0), (W_1, f_1), \dots)$, each $W_t$ consists only of features that can be deduced from $K_0$; that is, features that are definitely true based on $K_0$. Part 2 of the Proposition follows. □

*Proof of Proposition 5.* Under $C'$, given that $f$ is in $E'$, there exists some feasible working memory sequence $((W_0, f_0), \dots, (W_t, f_t))$ where $f = f_t$ is extrapolated for the first time in period $t$. Given that $C'$ and $C$ are identical outside of $q$, it follows that $\{(W_0, f_0), \dots, (W_t, f_t)\}$ is also feasible under $C$, and thus $f$ is also in $E$.

Consider the following example. Suppose $L = 4$, $\alpha = 1$, and $P$ is the $2 \times 1$ picture consisting of two white pixels. Let $q$ be the $2 \times 1$ pattern that exactly identifies $P$. Suppose $q$'s codeword in $C$ has length 2 and $q$'s codeword in $C'$ has length 3. Given that $\alpha = 1$, $f = (q, R)$ can only be extrapolated from $W_t$ if it can be deduced from $W_t$. But this would require that both pixels of the picture, as well as $f$, are loaded into working memory. This is feasible under $C$ (where the working memory requirement would be four) but not under $C'$ (where the working memory requirement would be five). It follows that $f \notin E'$ and $f \in E$. □

For the proofs of Propositions 6, 7, and 8, we will appeal to the setting described in the following Lemma. This setting is applicable to the baseline extrapolation model of Section 3, as well as the model of Section 4 where deletions are possible.

**Lemma 3.** *Let Picture $P$ be the $3 \times 3$ "checkerboard" matrix where pixel $p_{ij}$ is black if $i + j$ is even and white if $i + j$ is odd (see Figure 19). Let $K_0$ be the set of all nine pixels of $P$. Define the following $3 \times 3$ patterns. Pattern $q_r$ consists of all $3 \times 3$ matrices with at most one white pixel. Pattern $q'_r$ consists of all $3 \times 3$ matrices where the two leftmost columns have at most one white pixel. Pattern $q_d$ consists of all $3 \times 3$ matrices with at most one black pixel. Pattern $q'_d$ consists of all $3 \times 3$ matrices where the two leftmost columns have at most one black pixel. Denote the features corresponding to $q_r$, $q'_r$, $q_d$, $q'_d$ by $f_r$, $f'_r$, $f_d$, $f'_d$ respectively. Suppose that, under code $C$, patterns $q_r$, $q'_r$, $q_d$, and $q'_d$ have codewords of length two, and that no other patterns (other than pixels) have codewords of length three or less. Finally, suppose that $L = 4$ and $\alpha = 0.3$.*

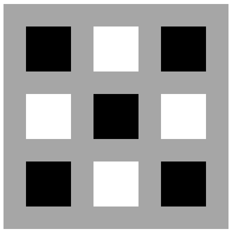


Figure 19: Picture $P$

1. *The agent cannot extrapolate to $f_r$, $f'_r$, $f_d$, or $f'_d$ from a single pixel (and nothing else) in working memory.*

2. *The agent cannot extrapolate to $f_r$ or $f'_r$ when either $f_d$ or $f'_d$ is in working memory, or vice versa.*

3. *Let $W_b$ ($W_w$) consist of any two of the black (white) pixels in P's two leftmost columns. The agent can extrapolate to $f_r$ and to $f'_r$ by evaluating these features against the contents of $W_b$. Analogously, the agent can extrapolate to $f_d$ and to $f'_d$ by evaluating these features against the contents of $W_w$.*

4. *When evaluating one of the features $f_r, f'_r, f_d$, or $f'_d$ against a single pixel, fit is strictly positive.*

5. *When evaluating either $f_r$ or $f'_r$ against $W_w$, or when evaluating either $f_d$ or $f'_d$ against $W_b$, fit equals zero.*

6. *When evaluating any $3 \times 3$ feature, both $f_r$ and $f_d$ are strictly more powerful features than $f'_r$ or $f'_d$.*

7. *Let $F' = \{f_r, f'_r, f_d, f'_d\}$. When evaluating any $3 \times 3$ feature, there is no feature (or set of features) not in $F'$ that is both simpler and more powerful—with at least one inequality strict—than any $f' \in F'$. Further, each feature $f' \in F'$ strictly dominates any other feature (except the $1 \times 1$ features and the features in $F'$), and also strictly dominates any set of two or more features.*

*Proof.*

1. Restrict attention to $f_r$ and $f'_r$ (without loss). Feature $f_r$'s fit when evaluated against a single black pixel is $\approx 0.17$. Feature $f'_r$'s fit when evaluated against a single black pixel in one of the two leftmost columns is $\approx 0.29$. These are the fit-maximizing choices for evaluating $f_r$ and $f'_r$ against a single pixel; thus the threshold $\alpha = 0.3$ cannot be met.

2. Notice that $f'_d$ implies that there must be at least 5 white pixels in the two leftmost columns of $P$; this is clearly inconsistent with both $f_r$ and $f'_r$. It follows that evaluating $f_r$ or $f'_r$ against $f_d$ or $f'_d$—or vice versa—leads to a fit of zero.

3. When evaluating against $W_b$, fit is $1/3$ in the case of $f_r$ and $\approx 0.54$ in the case of $f'_r$, which exceeds the threshold $\alpha = 0.3$ in both cases. The case of evaluating $f_d$ or $f'_d$ against $W_w$ is symmetric.

4. At least one matrix in each of $f_r$, $f'_r$, $f_d$, and $f'_d$ is consistent with any single pixel; thus fit when evaluating against any single pixel is strictly positive.

5. This follows immediately from the definitions of $f_r$, $f'_r$ and $W_w$.

6. Let $R$ be the region covering the entire $3 \times 3$ matrix. Then $Z(\{f_r\}, R) = 10$; there is one $3 \times 3$ matrix with zero white pixels and nine $3 \times 3$ matrices with one white pixel. Symmetrically, $Z(\{f_d\}, R) = 10$. We may also show that $Z(\{f'_r\}, R) = Z(\{f'_d\}, R) = 56$. The claim follows immediately.

7. The only possible sets of features that are weakly simpler than any feature in $F'$ are those sets consisting of (i) one $1 \times 1$ feature, or (ii) two $1 \times 1$ features. In both cases, it is easy to check that each feature in $F'$ is strictly more powerful than such a set.

□

*Proof of Proposition 6.* Consider the setting of Lemma 3. We know from Lemma 3 that the agent can extrapolate to $f_r$ if $f_d$ is not yet in knowledge (claim 3). However, once $f_d$ is in knowledge, the agent cannot extrapolate to $f_r$ because Assumption 1 ensures that $f_d$ will be prioritized in working memory over any other feature or set of features (claim 7), leading to a fit of zero (claim 2). Similarly, once $f_r$ is in knowledge, the agent cannot extrapolate to $f_d$. The result follows. □

*Proof of Proposition 7.* Consider the setting of Lemma 3 and suppose further that $\beta = 0$. We can rely on the claims in Lemma 3 to construct the following feasible sequence of events: (i) add $f_d$ to knowledge (claim 3); (ii) delete $f_d$ from knowledge (claim 5); (iii) add $f_r$ to knowledge (claim 3); delete $f_r$ from knowledge (claim 5); and finally repeat (i) and (ii). Such a sequence satisfies the cycling conditions specified in the proposition. □

*Proof of Proposition 8.* Consider the setting of Lemma 3 and suppose further that $\beta = 0$.

Given a feasible working memory sequence, suppose $f_r$ or $f_r'$ is in knowledge set $K_t$. We claim that neither $f_d$ nor $f_d'$ can be in $K_t$. For $\tau \in \{0, \dots t\}$, let $S_\tau$ be the statement "either $f_r$ or $f_r'$, and either $f_d$ or $f_d'$, are in $K_\tau$." Note that $S_0$ is obviously false. Assume towards a contradiction that $S_t$ is true. Let $t' = \max\{\tau : S_\tau \text{ is false}\}$, so that $S_{t'}$ is false and $S_{t'+1}$ is true. Then in period $t'$, one of the following events must have occured: either (i) one of $f_d$ and $f_d'$ was added to knowledge while $f_r$ or $f_r'$ was already in knowledge; or (ii) one of $f_r$ and $f_r'$ was added to knowledge while $f_d$ or $f_d'$ was already in knowledge. But, in light of Assumption 1, both cases contradict claims 2 and 7 of Lemma 3. Thus the claim holds.

Suppose $f_r$ and $f_r'$ are both in the knowledge set $K_t$, so that—by our claim above—neither $f_d$ nor $f_d'$ are in $K_t$. Then $f_r$ can never be subsequently deleted: given Assumption 1, claim 7 of Lemma 3 ensures that $f_r$ can only ever be evaluated against either a single pixel or against $f_r'$. In both cases, fit exceeds the threshold $\beta = 0$, so $f_r$ is not deleted. Similarly, $f_r'$ will never be deleted.

Next, suppose $f_r$ but not $f_r'$ is in the knowledge set $K_t$, so that—again by our claim above—neither $f_d$ nor $f_d'$ are in $K_t$. Then $f_r$ can be deleted in the next period by evaluating it against $W_w$ (claim 5 of Lemma 3). Similarly, $f_r'$ can be deleted if $f_r$ is not in the knowledge set. This establishes the result with $f = f_r$ and $f' = f_r'$. □

*Proof of Proposition 9.* Consider the following setting. $P$ is a $2 \times 1$ picture where the top pixel is black and the bottom pixel is white; $q_1$ is a $2 \times 1$ pattern "there is exactly one white pixel"; $q_2$ is a $2 \times 1$ pattern "the picture is either all black or all white"; and $f_1$ and $f_2$ are the corresponding features (see Figure 20). Each of $q_1$ and $q_2$ have length-two codewords; no other patterns have length-two codewords; and $K_0$ is the set of both pixels of $P$. Suppose $\alpha = 0.01$, $\beta = 0$, and the agent's working memory capacity is $L = 3$.

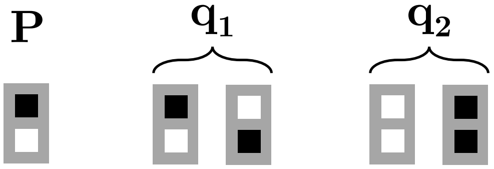


Figure 20: Proof of Patchwork Quilt

Consider the following sequence of extrapolations.

1. At $t = 0$, the agent evaluates $f_1$ by loading the bottom (white) pixel into working memory $W_0$. The fit of $f_1$ is then
$$\Phi(f_1, W_0) = 1/2,$$
which meets the threshold $\alpha = 0.01$; so $f_1$ is successfully added to the knowledge set.

2. At $t = 1$, the agent evaluates $f_2$, again by loading the bottom (white) pixel into working memory $W_1$. (Note that Assumption 1 has no bite here, because no feature is simpler than the single pixel.) The fit of $f_2$ is then
$$\Phi(f_2, W_1) = 1/2,$$
which again meets the threshold $\alpha = 0.01$; so $f_2$ is added to the knowledge set.

At the end of $t = 1$, $f_1$ and $f_2$ are both in the knowledge set. We claim that they will both remain forever in the knowledge set. To see why, note that the two codewords for $f_1$ and $f_2$ have length two, and thus (given $L = 3$) $f_1$ and $f_2$ can each only be removed by evaluation against a single pixel with a length-one codeword. However, we know from above that the fit from such an evaluation would be $1/2$; given that $\beta = 0$, neither feature will ever be deleted following such an evaluation. □

## A.2 Choice Problem: Analysis

This section analyzes the choice problem described in Section 5 and derives formulas for parameters $\theta$ and $\eta$.

*Selective Attention.* Suppose the agent considers whether to add $f_i$ to knowledge (i.e. choose option $i$), loading $a_1$ and $a_2$ into working memory (which we denote, in a slight abuse of notation, as $W_t = \{a_1, a_2\}$). To determine whether the agent adds $f_i$, we need to calculate the fit of $f_i$ given that $W_t = \{a_1, a_2\}$.

Let $X \equiv a_i - a_j$ denote the apple-difference between options $i$ and $j$. Let $R$ denote the region covering the entire picture. Observe that $f_i$ is a feature of region $R$, as it involves all four pixels of $P$ (it corresponds to the event $u(a_i, o_i) \geq u(a_j, o_j)$).

Observe that:

- $Z(\varnothing, R) = (z+1)^4$: when there are no "facts" in working memory ($W_t = \varnothing$), each of the four pixels of $P$ can take $z+1$ values (any integer value between 0 and $z$).
- $Z(W_t, R) = (z+1)^2$: when the values of two pixels ($a_1$ and $a_2$) are loaded into working memory, there are two remaining pixels that can take $z+1$ values.
- $Z(W_t \cup \{f_i\}, R) = |\{(o_1, o_2) : o_j \leq o_i + X\}|$, which is increasing in $X$.

Together, these results imply that $\Phi(f_i, W_t) = \frac{\log(Z(\varnothing),R) - \log(Z(W_t),R)}{\log(Z(\varnothing,R)) - \log(Z(W_t \cup \{f\},R))}$ is increasing in $X$. Given that the agent adds $f_i$ to knowledge (i.e. chooses option $i$) if and only if $\Phi(f_i, W_t) \geq \alpha$, it follows that the agent adds $f_i$ to knowledge if and only if $X$ exceeds some threshold $\theta$, where $\theta$ is the smallest value of $X$ that satisfies $\Phi(f_i, W_t) \geq \alpha$.

*Satisficing.* Next, suppose the agent considers whether to add $f_i$ to knowledge (i.e. choose option $i$), loading $a_i$ and $o_i$ into working memory ($W_t = \{a_i, o_i\}$). To determine whether the agent adds $f_i$, we need to calculate the fit of $f_i$ given that $W_t = \{a_i, o_i\}$. Let $Y \equiv a_i + o_i$ denote the utility associated with option $i$.

Observe that, similar to the selective attention case, $Z(\varnothing, R) = (z+1)^4$ and $Z(W_t, R) = (z+1)^2$. Furthermore: $Z(W_t \cup \{f_i\}) = |\{(a_j, o_j) : a_j + o_j \leq Y\}|$ is increasing in $Y$.

Together, these results imply that $\Phi(f_i, W_t) = \frac{\log(Z(\varnothing),R) - \log(Z(W_t),R)}{\log(Z(\varnothing,R)) - \log(Z(W_t \cup \{f\},R))}$ is increasing in $Y$. Given that the agent adds $f_i$ to knowledge (i.e. chooses option $i$) if and only if $\Phi(f_i, W_t) \geq \alpha$, it follows that the agent adds $f_i$ to knowledge if and only if $Z$ exceeds some threshold $\eta$, where $\eta$ is the smallest value of $Y$ that satisfies $\Phi(f_i, W_t) \geq \alpha$.

Suppose instead the agent attends to the attributes of option $j$ ($W_t = \{a_j, o_j\}$) when considering whether to add $f_i$ to knowledge (i.e. choose option $i$). Let $Y' \equiv a_j + o_j$. Then, observing that the expression $Z(W_t \cup \{f_i\}) = |\{(a_i, o_i) : a_i + o_i \geq Y'\}|$ is decreasing in $Y'$, it follows that the agent adds $f_i$ to knowledge if $Y'$ falls below some threshold $\eta'$.

*Complexity.* Finally, consider a setting where there are $n \geq 2$ options, with the $n = 3$ case illustrated below:

$$P = \begin{array}{c} \\ \text{apples} \\ \text{oranges} \end{array} \begin{array}{c} \begin{array}{ccc} \text{option 1} & \text{option 2} & \text{option 3} \end{array} \\ \left[ \begin{array}{ccc} a_1 & a_2 & a_3 \\ o_1 & o_2 & o_3 \end{array} \right] \end{array}.$$

Suppose the agent satisfices: they consider (without loss of generality) whether to add $f_1$ to knowledge (i.e. choose option 1), loading $a_1$ and $o_1$ into working memory ($W_t = \{a_1, o_1\}$). It is easy to verify, analogous to our calculations above, that the agent extrapolates to $f_1$ if and only if $Y = a_1 + o_1$ exceeds some threshold $\eta_n$. We claim that $\eta_n$ is increasing in $n$ (i.e. the satisficing threshold becomes harder to meet as the number of options grows).[49]

Define $Z_n(W_t, R)$ to be the value of $Z(W_t, R)$ given that there are $n$ options. Observe that $Z_n(\varnothing, R) = (z+1)^{2n}$ and $Z_n(W_t, R) = (z+1)^{2(n-1)}$; so that

$$\begin{aligned}\Phi(f_1, W_t) &= \tfrac{\log_{z+1}(Z_n(\varnothing),R) - \log_{z+1}(Z_n(W_t),R)}{\log_{z+1}(Z_n(\varnothing,R)) - \log_{z+1}(Z_n(W_t \cup \{f_1\},R))} \\ &= \tfrac{2}{2n - \log_{z+1}(Z_n(W_t \cup \{f_1\},R))}.\end{aligned}$$

Observe that fixing $n$, $Z_n(W_t \cup \{f_1\}, R)$ and thus $\Phi(f_1, W_t)$ are both increasing in $Y$. Consequently, to establish the claim, it is sufficient to show that for fixed $Y$, $\Phi(f_1, W_t)$ is decreasing in $n$; equivalently, that $\frac{Z_{n+1}(W_t \cup \{f_i\})}{Z_n(W_t \cup \{f_i\})} < (z+1)^2$ for each $n \geq 1$.

Recall that $Z_{n+1}(W_t \cup \{f_i\})$ is the number of attribute combinations $(a_2, o_2, \ldots, a_{n+1}, o_{n+1})$ that satisfy the linear constraints $a_j + o_j \leq Y, j \in \{2, \ldots, n+1\}$. Notice that for any combination that is counted in $Z_{n+1}(W_t \cup \{f_i\})$, the subsequence $(a_2, o_2, \ldots, a_n, o_n)$ must also satisfy $a_j + o_j \leq Y, j \in \{2, \ldots, n\}$, and thus must be counted in $Z_n(W_t \cup \{f_i\})$. For each such subsequence, there are at most $(z+1)^2$ possible pairs $(a_{n+1}, o_{n+1})$ to combine with; it follows that $\frac{Z_{n+1}(W_t \cup \{f_i\})}{Z_n(W_t \cup \{f_i\})} \leq (z+1)^2$. In fact, this inequality must be strict whenever $Y < 2z$, because any attribute combination $(a_2, o_2, \ldots, a_{n+1}, o_{n+1})$ with $(a_{n+1}, o_{n+1}) = (z, z)$ would violate the linear constraints for $Z_{n+1}(W_t \cup \{f_i\})$. The claim thus holds.

## A.3 Persuasion: Analysis

### The Role of Timing

We will fill in some details of the "timing" model from Section 6. The picture of interest is a vector $V = [p_1\, p_2\, p_3\, \cdots\, p_n]$ of $n$ pixels with $p_i \in \{\text{good}, \text{bad}\}$ and $n > 4$. There are two relevant features of the full vector: "good egg" where there are at most two bad pixels and "bad egg" where there are at most two good pixels.

There is a representative voter learning about $V$. The voter extrapolates as in Section 4.3, with addition and deletion thresholds $\alpha$ and $\beta$.

Only four pixels are ever revealed to the voter (i.e. are added exogenously to the knowledge set at some point). Two of these pixels are good and two are bad. These pixels are revealed in two rounds. The two good pixels or the two bad pixels are revealed in the first round; the remaining two pixels are revealed in the second round.

[49] We obtain an analogous result in the case of an agent who engages in selective attention (i.e. loads $a_1$ and $a_2$ into working memory to evaluate option $i$). Suppose we add attributes (e.g. bananas) and assume utility is additive across attributes (i.e. $u(a, o, b) = a + o + b$, where $b$ denotes bananas). In this case, the threshold $\theta$ is increasing in the number of attributes.

In each round, after pixels are revealed, the voter has many (but finitely many) periods, $t \in \{0, 1, 2, \ldots, T\}$, to make extrapolations. We assume that the voter undertakes a "rich" sequence of extrapolations in each period, in the following sense: given the agent's knowledge sequence in that round, $\{K_0, ..., K_T\}$, any other knowledge set $K'$ that could be attained by extrapolating from $K_T$ has been previously attained (i.e. $K' \in \{K_0, ..., K_T\}$).

We will work through an example with the following specific parameters, keeping in mind that the logic generalizes to a broader range of parameters. Assume that the voter has working memory capacity $L = 6$; each pixel's codeword has length one, "good egg" and "bad egg" are features with codewords of length two, and no other features have codewords of length six or less. Assume also that the picture has $n = 7$ pixels and that the extrapolation thresholds are $\alpha = 1/2$ for addition and $\beta = 1/100$ for deletion.

Assume, without loss of generality, that the two good pixels (collectively denoted "GG") are revealed to the voter in the first round, so that the two bad pixels are revealed in the second round.

*First Round.* The voter can extrapolate to "good egg" by loading both good pixels into memory:

$$\begin{aligned}\Phi(\text{good egg}, GG) &= \frac{\log(Z(\varnothing, R)) - \log(Z(GG, R))}{\log(Z(\varnothing, R)) - \log(Z(GG \cup \{\text{good egg}\}, R))} \\ &= \frac{2}{n - \log(1 + (n-2) + (n-2)(n-3)/2)} = 2/3 > \alpha\end{aligned}$$

where $R$ is the region covering the entire vector $V$. However, the voter cannot extrapolate to "bad egg":

$$\begin{aligned}\Phi(\text{bad egg}, GG) &= \frac{\log(Z(\varnothing, R)) - \log(Z(GG, R))}{\log(Z(\varnothing, R)) - \log(Z(GG \cup \{\text{bad egg}\}, R))} \\ &= \frac{2}{n - \log(1)} = 2/7 < \alpha.\end{aligned}$$

We can also easily check that the voter cannot extrapolate to any pixel $p$ that wasn't revealed to them, because none of the revealed pixels or the "good egg" feature serve to pin down $p$:

$$\Phi(p, \text{good egg}) = \frac{\log(Z(\varnothing, R_p)) - \log(Z(\text{good egg}, R_p))}{\log(Z(\varnothing, R_p)) - \log(Z(\text{good egg} \cup p, R_p))} = 0.$$

Given our assumption that the voter undertakes a "rich" set of extrapolations, they must have "good egg" but not "bad egg" in knowledge at the end of the first period.

*Second Round.* At the start of the second round, the voter has two bad pixels, two good pixels, and "good egg" (which they extrapolated to in the first round) in knowledge. If the voter only has pixels in working memory, "good egg" cannot be deleted from knowledge given the low threshold $\beta$; for instance, with both bad pixels in working memory,

$$\begin{aligned}\Phi(\text{good egg}, BB) &= \frac{\log(Z(\varnothing, R)) - \log(Z(BB, R))}{\log(Z(\varnothing, R)) - \log(Z(BB \cup \{\text{good egg}\}, R))} \\ &= \frac{2}{n - \log(1)} = 2/7 > \beta.\end{aligned}$$

Furthermore, Assumption 1 ensures that the voter cannot load two or more pixels into working memory to evaluate "bad egg" without also loading "good egg" into working memory. ("Good egg" takes up as much working memory as any two pixels and is strictly more powerful: there are 32 possible pictures given those two pixels, while there are only 29 possible pictures given "good egg".) Given that "good egg" and "bad egg" are mutually exclusive, the voter cannot extrapolate to "bad egg" by evaluating it against the two bad pixels (and "good egg"). In addition, the voter cannot extrapolate to "bad egg" by evaluating it against a single bad pixel (denoted $B$). In that case:

$$\begin{aligned}\Phi(\text{bad egg}, B) &= \frac{\log(Z(\varnothing, R)) - \log(Z(B, R))}{\log(Z(\varnothing, R)) - \log(Z(B \cup \{\text{bad egg}\}, R))} \\ &= \frac{1}{n - \log_2(n + (n-1)(n-2)/2)} = 1/(7 - \log_2(22)) \approx 0.39 < \alpha.\end{aligned}$$

It follows that the voter's knowledge set remains unchanged at the end of the second round. Since they think that candidate A is a "good egg," they vote for candidate A.

The case where the two bad pixels are revealed in the first round is symmetric. The voter extrapolates to "bad egg" in the first round; their knowledge set remains unchanged in the second round; and they vote for candidate B.

**Suggesting Narratives**

We now turn to the "suggesting narratives" model. We adopt the same vector of pixels $V$ as in the "timing" model. There is a particular subset of four pixels, two good and two bad, that we label the "story." Assume that the agent has the same working memory capacity, code, and thresholds $\alpha$ and $\beta$ as they did in our example from the "timing" model—only with features "forgivable" and "unforgivable" in place of "good egg" and "bad egg."

The voter knows none of the features of $V$ at the start of the game. Candidates $A$ and $B$ take turns influencing the voter, with candidate $A$ moving first, as follows. At the start of their turn, candidate A decides whether to reveal the story to the voter (i.e. add the four pixels to knowledge). At the start of their turn, candidate B decides whether to reveal the story *if they are aware of the story (and the story has not been revealed already).*

On each candidate's turn, following their revelation decision, the voter engages in a "rich" sequence of extrapolations over multiple periods, with corresponding knowledge sequence $\{K_0, ..., K_T\}$. In periods $t = 0$ and $t = 1$, the active candidate can choose the facts ($W_0$ and $W_1$) that the voter loads into working memory and the features ($f_0$ and $f_1$) that the voter evaluates against those facts—subject to Assumption 1.

Recall that if the voter knows the "forgivable" feature, they choose candidate $A$ with probability $p$; if they know the “unforgivable” feature, they choose candidate $A$ with probability zero; and if they know neither feature, they choose candidate $A$ with probability 1. Recall also that candidate $B$ is aware of the story with probability $q$.

Consider candidate $A$'s decision about whether to reveal the story. Suppose candidate $A$ reveals the story. Our analysis from the “timing” model tells us that candidate $A$ can, in period 0 of their turn, get the voter to consider the “forgivable” narrative ($f =$”forgivable”) while focusing the voter's attention on the good pixels ($W_0 = \{GG\}$). The voter will adopt the “forgivable” narrative; moreover, no further changes can be made to the voter's knowledge set on candidate $B$'s turn.

Notice also that if candidate $A$ reveals the story but does not get the voter to consider/adopt the “forgivable” narrative, candidate $B$ can (and will) get the voter to consider the “unforgivable” narrative—along with $BB$—on their turn. The voter then adopts the “unforgivable” narrative and no further changes can be made to the voter's knowledge set.

It is thus optimal (conditional on the story being revealed) for candidate $A$ to add “forgivable” to knowledge, in which case candidate $A$ wins with probability $q$.

Suppose now that candidate $A$ does not reveal the story. If candidate $B$ is aware of the story (which occurs with probability $p$), they will reveal the story and add “unforgivable” to the voter's knowledge set. Otherwise, candidate $B$ can do nothing, and the voter has neither “forgivable” nor “unforgivable” in knowledge. Thus, candidate $A$ wins with probability $1 - p$ and loses with probability $p$.

Comparing these outcomes, candidate $A$ prefers to reveal the story if and only if $q > 1 - p$ (i.e. $p + q > 1$).

# References

**Arpan, Laura M and David R Roskos-Ewoldsen**, "Stealing thunder: Analysis of the effects of proactive disclosure of crisis information," *Public Relations Review*, 2005, *31* (3), 425–433.

**Asch, Solomon E**, "Forming impressions of personality.," *The journal of abnormal and social psychology*, 1946, *41* (3), 258.

**Baddeley, A D and G Hitch**, "Working Memory," in G H Bower, ed., *The psychology of learning and motivation: Advances in research and theory*, Vol. 8, New York: Academic Press, 1974, pp. 47–89.

**Baumeister, RF, E Bratslavsky, M Muraven, and DM Tice**, "Ego depletion: is the active self a limited resource?," *Journal of Personality and Social Psychology*, 1998, *74* (5), 1252–1265.

**Bénabou, Roland, Armin Falk, and Jean Tirole**, "Narratives, imperatives, and moral reasoning," *NBER Working Paper 24798*, 2018.

**Bordalo, Pedro, John J Conlon, Nicola Gennaioli, Spencer Y Kwon, and Andrei Shleifer**, "Memory and probability," *Quarterly Journal of Economics*, 2023, *138* (1), 265–311.

**_ , Katherine Coffman, Nicola Gennaioli, and Andrei Shleifer**, "Stereotypes," *Quarterly Journal of Economics*, 2016, *131* (4), 1753–1794.

**_ , Nicola Gennaioli, and Andrei Shleifer**, "Salience theory of choice under risk," *Quarterly Journal of Economics*, 2012, *127* (3), 1243–1285.

**_ , _ , and _ ,** "Salience and asset prices," *American Economic Review*, 2013, *103* (3), 623–628.

**_ , _ , and _ ,** "Salience and consumer choice," *Journal of Political Economy*, 2013, *121* (5), 803–843.

**_ , _ , and _ ,** "Competition for attention," *Review of Economic Studies*, 2016, *83* (2), 481–513.

**_ , _ , and _ ,** "Memory, attention, and choice," *Quarterly Journal of Economics*, 2020, *135* (3), 1399–1442.

**_ , _ , Giacomo Lanzani, and Andrei Shleifer**, "A Cognitive Theory of Reasoning and Choice," 2024.

**Chang, Ruth**, "Hard choices," *Journal of the American Philosophical Association*, 2017, *3* (1), 1–21.

**Chase, William G and Herbert A Simon**, "Perception in chess," *Cognitive Psychology*, 1973, *4* (1), 55–81.

**Chater, Nick**, "Reconciling simplicity and likelihood principles in perceptual organization.," *Psychological review*, 1996, *103* (3), 566.

_ **and Paul Vitányi**, "Simplicity: a unifying principle in cognitive science?," *Trends in cognitive sciences*, 2003, *7* (1), 19–22.

**Craik, Fergus IM and Robert S Lockhart**, "Levels of processing: A framework for memory research," *Journal of Verbal Learning and Verbal Behavior*, 1972, *11* (6), 671–684.

**Crawford, Vincent P. and Nagore Iriberri**, "Level-K Auctions: Can a Nonequilibrium Model of Strategic Thinking Explain the Winner's Curse and Overbidding in Private-Value Auctions?," *Econometrica*, 2007, *75* (6), 1721–1770.

**Cremer, Jacques, Luis Garicano, and Andrea Prat**, "Language and the Theory of the Firm," *Quarterly Journal of Economics*, 2007, *122* (1), 373–407.

**Davidson, Donald**, "Actions, reasons, and causes," *Journal of Philosophy*, 1963, *60*, 685–700.

**De Groot, A D**, *Thought and choice in chess*, Mouton, 1965.

**Dhar, Ravi**, "Consumer preference for a no-choice option," *Journal of Consumer Research*, 1997, *24* (2), 215–231.

**Eliaz, Kfir and Ran Spiegler**, "A model of competing narratives," *American Economic Review*, 2020, *110* (12), 3786–3816.

**Ellison, Glenn and Richard Holden**, "A theory of rule development," *The Journal of Law, Economics, & Organization*, 2014, *30* (4), 649–682.

**Enke, Benjamin and Thomas Graeber**, "Cognitive uncertainty," *Quarterly Journal of Economics*, 2023, *138* (4), 2021–2067.

**Entman, Robert M**, "Framing: Toward clarification of a fractured paradigm," *Journal of communication*, 1993, *43* (4), 51–58.

**Feldman, Jacob**, "Minimization of Boolean complexity in human concept learning," *Nature*, 2000, *407* (6804), 630–633.

**Gabaix, Xavier**, "A sparsity-based model of bounded rationality," *Quarterly Journal of Economics*, 2014, *129* (4), 1661–1710.

**Gazzaniga, Michael S., Richard B. Ivry, and George R. Mangun**, "Cognitive neuroscience: the biology of the mind," 2014.

**Gibbons, Robert, Marco LiCalzi, and Massimo Warglien**, "What situation is this? Shared frames and collective performance," *Strategy Science*, 2021, *6* (2), 124–140.

**Gilbert, Charles D and Wu Li**, "Top-down influences on visual processing," *Nature Reviews Neuroscience*, 2013, *14* (5), 350–363.

**Gregory, Richard Langton**, *The intelligent eye.*, Littlehampton, 1970.

**Grünwald, Peter D**, *The minimum description length principle*, MIT press, 2007.

**Iyengar, Sheena S and Mark R Lepper**, "When choice is demotivating: Can one desire too much of a good thing?," *Journal of Personality and Social Psychology*, 2000, *79* (6), 995.

**Johnson, Samuel GB, Avri Bilovich, and David Tuckett**, "Conviction narrative theory: A theory of choice under radical uncertainty," *Behavioral and Brain Sciences*, 2023, *46*, 1–26.

**Kamenica, Emir and Matthew Gentzkow**, "Bayesian persuasion," *American Economic Review*, 2011, *101* (6), 2590–2615.

**Lakoff, George**, *The all new don't think of an elephant!: Know your values and frame the debate*, Chelsea Green Publishing, 2014.

**Leopold, David A and Nikos K Logothetis**, "Multistable phenomena: changing views in perception," *Trends in cognitive sciences*, 1999, *3* (7), 254–264.

**Luntz, Frank**, *Words That Work: It's Not What You Say, It's What People Hear*, Hachette UK, 2007.

**Madrian, Brigitte C and Dennis F Shea**, "The power of suggestion: Inertia in 401 (k) participation and savings behavior," *Quarterly Journal of Economics*, 2001, *116* (4), 1149–1187.

**Marr, David**, *Vision: A computational investigation into the human representation and processing of visual information*, MIT press, 2010.

**McAdams, Dan P**, *The stories we live by: Personal myths and the making of the self*, William Morrow, 1993.

**Medin, Douglas L and Marguerite M Schaffer**, "Context theory of classification learning.," *Psychological review*, 1978, *85* (3), 207.

**Miller, George A**, "The magical number seven, plus or minus two: Some limits on our capacity for processing information.," *Psychological Review*, 1956, *63* (2), 81.

**Mullainathan, Sendhil**, "A memory-based model of bounded rationality," *Quarterly Journal of Economics*, 2002, *117* (3), 735–774.

**_ , Joshua Schwartzstein, and Andrei Shleifer**, "Coarse thinking and persuasion," *Quarterly Journal of Economics*, 2008, *123* (2), 577–619.

**Neisser, Ulric**, *Cognitive psychology: Classic edition*, Psychology press, 2014.

**Oaksford, Mike and Nick Chater**, *Bayesian rationality: The probabilistic approach to human reasoning*, Oxford University Press, 2007.

**Payne, John W, James R Bettman, and Eric J Johnson**, *The Adaptive Decision Maker*, Cambridge University Press, 1993.

**Ricoeur, Paul**, *Oneself as another*, The University of Chicago Press, 1992.

**Rissanen, Jorma**, "Modeling by shortest data description," *Automatica*, 1978, *14* (5), 465–471.

**Rosch, Eleanor H**, "Natural categories," *Cognitive Psychology*, 1973, *4* (3), 328–350.

**Sah, Raaj Kumar and Joseph E Stiglitz**, "The architecture of economic systems: Hierarchies and polyarchies," *American Economic Review*, 1986, *76* (4), 716–727.

**Schwartzstein, Joshua and Adi Sunderam**, "Using models to persuade," *American Economic Review*, 2021, *111* (1), 276–323.

**Serences, John T and Steven Yantis**, "Selective visual attention and perceptual coherence," *Trends in cognitive sciences*, 2006, *10* (1), 38–45.

**Shafir, Eldar, Itamar Simonson, and Amos Tversky**, "Reason-based choice," *Cognition*, 1993, *49* (1-2), 11–36.

**Shiller, Robert J**, "Narrative economics," *American Economic Review*, 2017, *107* (4), 967–1004.

**Simon, Herbert A**, *Administrative behavior*, Simon and Schuster, 1947.

_ , "A behavioral model of rational choice," *Quarterly Journal of Economics*, 1955, pp. 99–118.

**Sims, Christopher A**, "Implications of rational inattention," *Journal of Monetary Economics*, 2003, *50* (3), 665–690.

**Stahl, Dale O. and Paul W. Wilson**, "Experimental evidence on players' models of other players," *Journal of Economic Behavior and Organizations*, 1994, *25* (3), 309–327.

**Sweller, John**, "Cognitive load during problem solving: Effects on learning," *Cognitive Science*, 1988, *12* (2), 257–285.

**Tenenbaum, Joshua B, Thomas L Griffiths, and Charles Kemp**, "Theory-based Bayesian models of inductive learning and reasoning," *Trends in cognitive sciences*, 2006, *10* (7), 309–318.

**Thaler, Richard H and Shlomo Benartzi**, "Save more tomorrow™: Using behavioral economics to increase employee saving," *Journal of Political Economy*, 2004, *112* (S1), S164–S187.

**Tinbergen, Nikolaas and Albert C Perdeck**, "On the stimulus situation releasing the begging response in the newly hatched herring gull chick (Larus argentatus argentatus Pont.)," *Behaviour*, 1951, *3* (1), 1–39.

**Tversky, Amos**, "Elimination by aspects: A theory of choice.," *Psychological review*, 1972, *79* (4), 281.

— **and Eldar Shafir**, "Choice under conflict: The dynamics of deferred decision," *Psychological Science*, 1992, *3* (6), 358–361.

**Wertheimer, Max**, "Laws of organization in perceptual forms," in W. D. Ellis, ed., *A Source Book of Gestalt Psychology*, Kegan Paul, Trench, Trubner & Co., 1938, pp. 71–88.

**Wilson, Andrea**, "Bounded memory and biases in information processing," *Econometrica*, 2014, *82* (6), 2257–2294.

**Wojtowicz, Zachary**, "Model Diversity and Dynamic Belief Formation," *Working paper*, 2024.